# Paired Comparison: A User Perspective

Shigeru Sasao
Master of Software Engineering
Carnegie Mellon University

ssasao@andrew.cmu.edu

**Abstract**

Paired comparison is an expert based judgment estimation technique. The method reduces individual judgment errors by requiring multiple pair-wise comparisons of relative sizes. Observations made in an academic environment shows that paired comparison produces more consistent results across estimators compared to ad-hoc methods and planning poker. A tool is introduced, which implements a more efficient version of the original paired comparison method which employs cyclic designs to reduce the number of comparisons required without a corresponding loss in reliability.

## I. Introduction

Paired comparison is an expert judgment estimation technique that uses a relative size comparison of multiple entities against one another to compensate for errors on individual judgments. Observations in an academic environment indicate that estimates from paired comparison produce a narrower spread of values, thus more repeatable estimations, as compared to ad-hoc estimation and planning poker. This result was consistent with previous reports. [Miranda 2000]

Several points of improvement were observed during the estimation process. First, comparison of a large set of components was difficult due to the amount of stamina required from the estimators. For this point, a solution is investigated to use a new version of paired comparison with lower replication factors to reduce the number of comparisons in the set. Secondly, as estimators became fatigued, they started extrapolating new judgments from previous judgments, thus weakening the compensatory effect of having multiple comparisons. To address this issue, randomization should be used to sequence the comparisons. This paper will present the results of experiments conducted by the author among his fellow students at the Master of Software Engineering program at Carnegie Mellon University to validate the method, and introduce a tool created by the author that implements the new version of paired comparison using cyclic designs. [Miranda et al. 2009]

## II. Introduction of Paired Comparison

Imagine that we have three components X, Y, and Z. Also, let us assume that we judge X to be twice as big as Y, and Z to be twice as big as X. If the size of X is 4 units, logically, we know the size of Y to be 2 units and Z to be 8 units. So, instead of directly estimating the size of each component, we can reason about the relative sizes of the components, and extract the size estimates from these values. The

paired comparison method uses a relative size comparison of multiple entities against one another to compensate for errors on individual judgments.

For example, let us assume we have four programs {A, B, C, D}, where we know the actual size of C to be 10 Kilo Lines of Code (KLOC). We will use the information about the size of C later, after we generate the ratio indexes of each entity. Experts are asked to assess the relative sizes of the entities against each other. For example, an expert might say that program A is 3 times bigger than program D. By comparing all possible pairs, we can construct a judgment matrix as follows:

| | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| **A** | 1 | 0.5 | 2 | 3 |
| **B** | 2 | 1 | 0.5 | 6 |
| **C** | 0.5 | 2 | 1 | 2 |
| **D** | 0.3 | 0.17 | 0.5 | 1 |

**Figure 1**
Judgment Matrix

Notice that the relative values do not need to be consistent among rows, since it is an estimation. Also, notice that the principal diagonal is always one, and the other shaded areas are simply reciprocals of the unshaded region. To calculate the ratios from the relative sizes, we use the method proposed by Crawford and Williams [Crawford, Williams 1985] which is explained below. First, we need to calculate the geometric mean for each entity (program).

$$v_i = \sqrt[n]{\prod_{j=1}^{n} a_{ij}}$$

The geometric mean for the above example will be as follows:

| | |
|---|---|
| **A** | 1.32 |
| **B** | 1.57 |
| **C** | 1.19 |
| **D** | 0.41 |

**Table 1**
Geometric Mean of each row in the judgment matrix

The ratio scale can then be obtained using the following equation:

$$r_i = \frac{v_i}{\sum_{l=1}^{n} v_l}$$

For the above example, we will sum the geometric means,

$$1.32 + 1.57 + 1.19 + 0.41 = 4.49$$

and divide each geometric mean with the sum of the geometric means:

| | |
|---|---|
| **A** | 0.29 |
| **B** | 0.35 |
| **C** | 0.26 |
| **D** | 0.09 |

**Table 2**
Ratio scale

This is the normalized ratio scale for our paired comparison. Now, we know that the size of program C is 10 KLOC, which is our reference point. Therefore, we can use the ratio scale to obtain the estimated sizes of the remaining entities using the following equation:

$$Size_i = \frac{r_i}{r_{ref}} * Size_{ref}$$

C will be used to substitute for $Size_{ref}$. The below table shows how to calculate the estimated sizes for our example:

| | |
|---|---|
| **A** | 0.29/0.26 * 10 = 11.15 KLOC |
| **B** | 0.35/0.26 * 10 = 13.46 KLOC |
| **C** | 0.26/0.26 * 10 = 10.00 KLOC |
| **D** | 0.09/0.26 * 10 = 3.46 KLOC |

**Table 3**
Estimated size

One important point to consider about paired comparison is that the difference of entities being compared is within one order of magnitude. If one entity exists that is vastly larger or smaller than the rest, it could offset the entire estimation. For further discussions on the underlying theory behind paired comparison, please refer to [Miranda et al. 2009] and [Shepperd et al. 2001].

## III. Case Studies

Observations were made on the usage of paired comparison in a classroom environment and an academic project. First, a comparison was made between the use of paired comparison, planning poker and ad-hoc estimations in a classroom environment. Secondly, an observation was made on the use of paired comparison to estimate effort for architectural components in an academic project. Both studies were conducted among students from the Master of Software Engineering program at Carnegie Mellon, where students have a strong technical background with two to three years of industry experience.

### 1. Comparing Ad-hoc, Planning Poker and Paired Comparison Estimations

In a classroom, students were divided into three groups, with each group using either ad-hoc, planning poker, or paired comparison as estimation techniques. Estimations were conducted in pairs to

control for differences between individual and group estimates[1], with five pairs per estimation technique (five data points per estimation technique). Students were asked to estimate the size (LOC) of different data structures as described in Table 4. Prior to estimation, all students were told that the size of "linked list (a)" was 40 LOC to control for the use of a reference value, and were asked to estimate the remaining data structures. The size for "linked list (a)" was used as the reference value for the paired comparison. The data structures selected were purposefully the same as in [Miranda 2001] to examine whether the results were replicable.

| Data Structure | Definition |
|---|---|
| Stack | The stack program stores and returns simple numbers. The operations detect usual error conditions. It implements the following operations: Add, Delete, Top, IsEmpty. |
| Queue | The program stores and returns simple numbers. The operations detect usual error conditions. It implements the following operations: Add, Delete, First, IsEmpty. |
| Binary Tree | The binary tree program stores and returns simple numbers. The operations detect usual error conditions. It implements the following operations: Root, AddLeft, AddRight, TraverseLeft, TraverseRight, Delete, Data, IsEmpty. |
| Linked List (a) | The Linked List (a) program manipulates an ordered list of linked fixed length elements. The operations detect usual error conditions. It implements the following operations: First, Insert, Delete, Find, IsEmpty. |
| String Manipulation | This program manipulates a string of characters of arbitrary length. The operations detect usual error conditions. It implements the following operations: Null, IsNull, Length, Insert, Concatenate, Equal, Index, Substring. |
| Linked List (b) | The Linked List (b) program manipulates a list of variable length elements, and includes a garbage collection mechanism. The operations detect usual error conditions. It implements the following operations: Create, First, Insert, Delete, Find, IsEmpty. |
| Balanced Tree | This program implements the same operations as the Binary Tree Program, with the addition that none of the height of the branches could exceed the height of any other branch for more than 1. An imbalance will trigger the automatic reorganization of the whole tree. The reorganization shall be such as not to alter the traversal order of the tree. |
| Hash Table | The Hash Table program stores and returns character strings. The operations detect usual error conditions. It implements the following operations: Add, Delete, Find, IsEmpty. Freed space is recycled. |

**Table 4**
Students were asked to estimate the size of the above data structures.

Figure 2 shows a box plot of the resulting estimates. As observed from the size of the box plots, both planning poker and paired comparison show a vast improvement in precision over the ad-hoc estimation method. This result is consistent with previous reports from [Miranda 2000], which compared the accuracy and precision between ad-hoc methods and paired comparison. In addition, this experiment shows the comparison in precision between planning poker and paired comparison. For all

[1] While ad-hoc and paired comparison can be performed by a single estimator, planning poker requires at least two estimators.

seven data structures, Table 5 shows that paired comparison resulted in a lower standard deviation as compared to planning poker. This indicates that paired comparison produces more precise, thus more repeatable estimation results as compared to ad-hoc methods and planning poker.

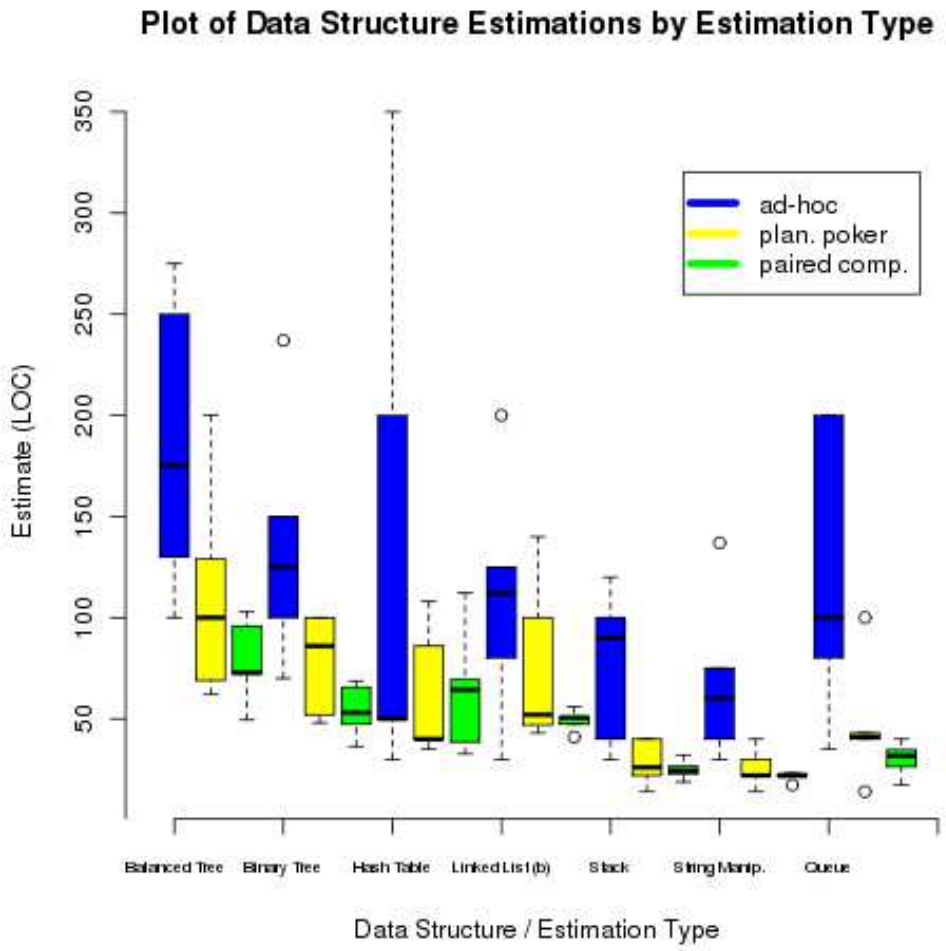


**Figure 2**
Box plot of results from ad-hoc (blue), planning poker (yellow) and paired comparison (green).

| Data Structure | Std. Dev. (Planning Poker) | Std. Dev. (Paired Comparison) |
|---|---|---|
| Stack | 9.8 | 2.4 |
| Queue | 11.4 | 4.9 |
| Binary Tree | 25.5 | 13.3 |
| String Manipulation | 31.6 | 8.6 |
| Linked List (b) | 42.4 | 5.5 |
| Balanced Tree | 56.0 | 21.4 |
| Hash Table | 33.1 | 31.5 |

**Table 5**
Comparison of standard deviation between planning poker and paired comparison.

## 2. Academic Project

As a part of the curriculum in the Master of Software Engineering (MSE) program, students work in a project for an external customer. The project runs for 16 months, with a team of four or five members for a total resource of 4608 to 5760 person hours. Observations were made on a MSE team that used paired comparison to estimate the required effort of the project. The project was to develop a globally integrated development environment for a large multinational corporation. The estimation was conducted after the architecture of the system was already finalized, prior to the detailed design and implementation stage. The team used paired comparison to directly estimate effort for the components in their architecture.

The five members of the team and the observers gathered in a conference room for the estimation session. A list of 12 components was prepared and entered into a spreadsheet tool in advance to document the comparisons. At the beginning of the meeting, a decision was made among the team whether to use a verbal scale such as “Slightly Bigger”, “Much Smaller” or to use numerical factors. [Miranda 2001] The team chose to use numerical factors.

The pair-wise comparison was conducted on each field of the judgment matrix. At each field, team members discussed issues such as size, complexity and number of state transitions in order to agree on a numerical comparison factor of pair-wise elements. Once this value was agreed upon, a team member entered the value into the original tool developed by Miranda as shown in Figure 3. [Miranda 2001] Then, the team selected the next comparison field in a left to right, top to bottom manner. In total, 66 pair-wise comparisons were made for the 12 components. The estimation session took a total of 54 minutes to compare the 12 elements

| Reference Value | Artifact Name | Data Elem. Mgr | Traceability controller | Import UI | Trac. View Cont | Data UI | Traceability UI | Traceability Server | ADT Def | Transf. Chain | Import Mgr | Traceability Conf | RTC Bridge | | | | Ratio Scale | Estimated Value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data Elem. Mgr | | 2.0 | 9.5 | 4.1 | 4.5 | 4.3 | 5.5 | 5.6 | 7.0 | 9.0 | 8.0 | 1.5 | | | | 0.256 | 258.7 |
| | Traceability controller | | | 7.0 | 1.8 | 2.5 | 2.2 | 2.0 | 2.2 | 5.5 | 6.5 | 6.0 | 0.5 | | | | 0.135 | 136.5 |
| 20 | Import UI | | | | 0.5 | 0.2 | 0.2 | 0.4 | 0.4 | 0.7 | 0.8 | 0.9 | 0.1 | | | | 0.020 | 20.0 |
| | Trac. View Cont | | | | | 2.0 | 1.8 | 1.5 | 1.6 | 2.2 | 3.5 | 3.0 | 0.3 | | | | 0.076 | 76.7 |
| | Data UI | | | | | | 0.8 | 0.5 | 0.6 | 0.8 | 2.8 | 2.4 | 0.4 | | | | 0.050 | 50.8 |
| | Traceability UI | | | | | | | 0.8 | 0.7 | 2.5 | 3.8 | 3.5 | 0.4 | | | | 0.067 | 67.4 |
| | Traceability Server | | | | | | | | 1.1 | 2.0 | 3.5 | 3.2 | 0.4 | | | | 0.067 | 67.9 |
| | ADT Def | | | | | | | | | 2.2 | 3.7 | 3.4 | 0.4 | | | | 0.067 | 67.5 |
| | Transf. Chain | | | | | | | | | | 2.5 | 2.8 | 0.3 | | | | 0.038 | 38.4 |
| | Import Mgr | | | | | | | | | | | 1.6 | 0.1 | | | | 0.021 | 21.0 |
| | Traceability Conf | | | | | | | | | | | | 0.1 | | | | 0.020 | 20.4 |
| | RTC Bridge | | | | | | | | | | | | | | | | 0.182 | 183.7 |

**Figure 3**

Snapshot of the completed judgment matrix by the MSE team, using an old version of the tool requiring every entity to be compared to every other entity.

## IV. The New Tool

In the above case study, two areas of improvement were observed for the paired comparison technique. First, comparison of a large set of components was difficult due to the estimators’ stamina. In the case study, the comparison of 12 data structures required a total of 66 comparisons. The estimators

became noticeably tired during the exercise. Secondly, as estimators became tired, they began extrapolating the new relative size values from old ones that have already been determined. Paired comparison uses multiple comparisons for each entity to reduce the error in judgment. Thus, by extrapolating the values from existing ones, the compensatory effect of the multiple comparisons was reduced.

Based on the observations above, the author developed a new tool to address these areas of improvements. In order to use paired comparison effectively while considering the estimators' stamina, the proposed solution is to use a new version of paired comparison with cyclic designs as introduced by Miranda. [Miranda et al. 2009] In this new version of paired comparison, estimators can reduce the number of comparisons for a fixed number of components by controlling a parameter called the replication factor. For example, a full comparison of 10 elements will require 45 comparisons. Table 6 shows how the number of comparison can be reduced by lowering the replication factor.

| Replication Factor | Number of Comparisons |
|---|---|
| 10 | 45 |
| 8 | 40 |
| 6 | 30 |
| 4 | 20 |
| 2 | 10 |

**Table 6**
Replication factor versus required number of comparisons for 10 components.

The reduction of comparison is realized through the use of incomplete cyclic design (ICD) and imputing the missing values in the judgment matrix as the geometric mean of the populated values of the row. For a further discussion on the ICD and imputation algorithm, please refer to [Spencer 1982] and [Miranda et al. 2009]. To further illustrate how the use of cyclic design reduces the number of comparisons, consider an estimation of 8 entities. Figure 4 shows the full comparison, with a total of 28 comparisons. By applying the replication factor of 4, the number of comparisons is reduced to 16 comparisons, as shown in Figure 5.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | | | | | | | | |
| **B** | | | | | | | | |
| **C** | | | | | | | | |
| **D** | | | | | | | | |
| **E** | | | | | | | | |
| **F** | | | | | | | | |
| **G** | | | | | | | | |
| **H** | | | | | | | | |

**Figure 4**
Full comparison for 8 elements (replication factor = 8), 28 comparisons.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | | | | | | | | |
| **B** | | | | | | | | |
| **C** | | | | | | | | |
| **D** | | | | | | | | |
| **E** | | | | | | | | |
| **F** | | | | | | | | |
| **G** | | | | | | | | |
| **H** | | | | | | | | |

**Figure 5**
Comparison with reduced replication factor for 8 elements (replication factor = 4), 16 comparisons.

Naturally, the reduction in the replication factor will reduce the compensatory effects of having multiple comparisons. Empirical research shows that very low replication factors still produce acceptable results in terms of the Mean Magnitude of Relative Error (MMRE), but the consistency of the estimators need to increase as the replication factor is decreased. In fact, having a replication factor of 2 reduces the paired comparison method to a "triangulation estimation"[2], and estimators will need to have almost perfect consistency. Estimators need to balance the reduction of effort from lower replication factors against the reduction of the compensatory effect of having more comparisons.

Estimators may think about reducing the number of comparison in one estimation session by dividing the set of comparisons into smaller sets. For example, a set of 20 components may be divided into 2 sets of 10 components, and paired comparison can be applied separately on each set. However, doing so will break the property of connectedness [Spencer 1982] of the experiment design. Connectedness for the experiment design is a property that every component in the set is directly or indirectly compared to every other component in the set. Studies have shown that expected performance of the ICD is related to its connectedness. By splitting the set, components in different subsets will no longer be compared and consistency will be lost between the subsets. For example, let us assume that we break a comparison set with elements {A, B, C, D, E, F} into two sets: {A, B, C} and {D, E, F}. In this case, elements in {A, B, C} will never be compared to {D, E, F}, thus estimations may be inconsistent between the two subsets.

As mentioned before, fatigued estimators began extrapolating the new comparison values from old ones, instead of discussing the relative sizes as they did at the beginning of the estimation session. To minimize this risk, the tool implemented by the author includes a mode by which the tool randomly selects the comparison to be made at a given point. Figure 6 shows a snapshot of the tool. The fields on the top left of the tool (number of components and replication factor) can be modified to control the number of comparisons. The tool also implements a "Guide" button (bottom left), which guides the estimators during the comparison by randomly selecting cells in the judgment matrix which are not yet compared

[2] Triangulation is the size estimation of an entity by comparing it to two other entities.

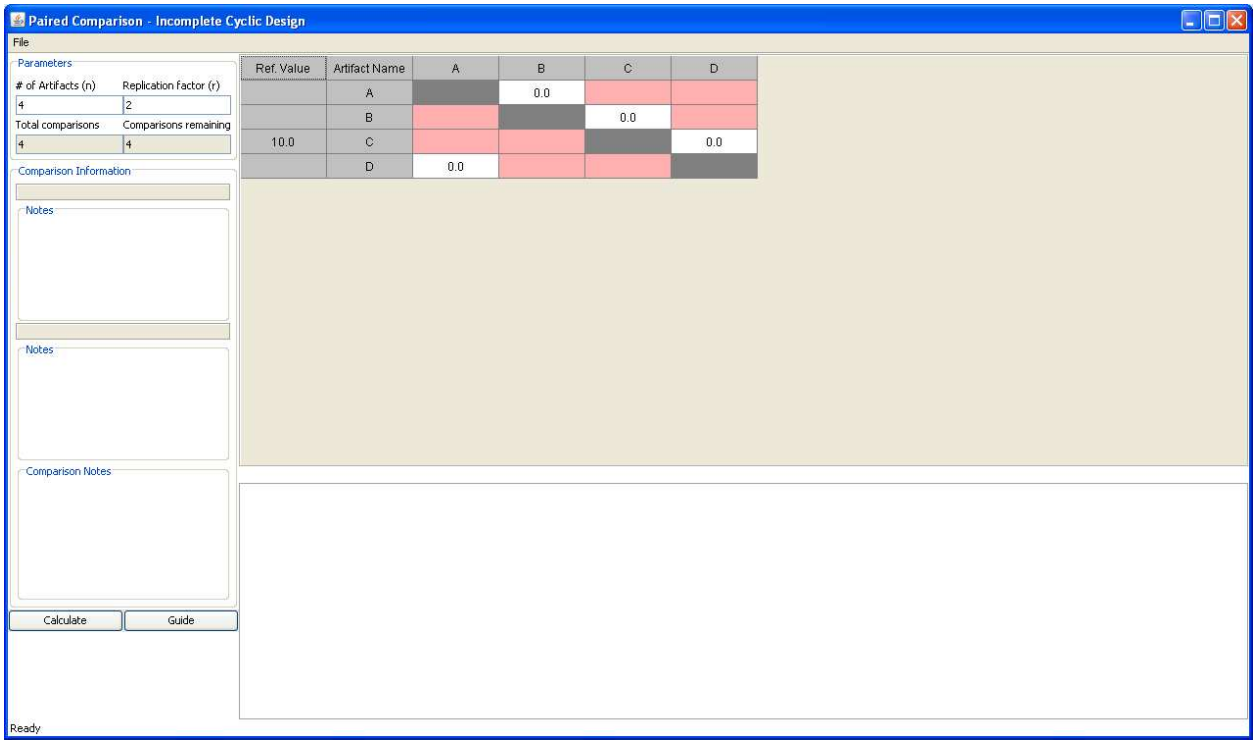


**Figure 6**
Paired Comparison tool developed by the author, which implements the new version of paired comparison with using cyclic design.

## V. Summary

The conducted experiment confirmed that paired comparison produces more consistent estimations than ad-hoc methods and planning poker. It can be used to estimate effort, or to estimate LOC size of software components as inputs into parametric models such as COCOMO. As a result of observations made during the conduct of the experiment and the usage of the tool, the author developed a new tool to improve on the scalability of the method. If you are interested in obtaining the tool, please contact the author.

**References**

[Aguaron et al. 2003] Aguaron, Juan, Moreno-Jimenez, Jose Maria, *The Geometric Consistency Index: Approximated Thresholds*, European Journal of Operational Research, 147 (2003), Pages 137-145.

[Crawford, Williams 1985] Crawford, Gordon, Williams, Cindy, *The Analysis of Subjective Judgment Matrices*, Rand Corporation, 1985.

[Miranda 2000] Miranda, Eduardo, *An Evaluation of the Paired Comparisons Method for Software Sizing*, Proceedings of the 22nd International Conference on Software Engineering, 2000.

[Miranda 2001] Miranda, Eduardo, *Improving Subjective Estimates Using Paired Comparison*, IEEE Software, 2001.

[Miranda et al. 2009] Miranda, Eduardo, Bourque, Pierre, Abran, Alain, *Sizing User Stories Using Paired Comparisons*, Information and Software Technology Volume 51, Issue 9, September 2009, Pages 1327-1337, Butterworth-Heinemann, 2009.

[PlanningPoker] Cohn, Mike, *Planning Poker*, www.planningpoker.com, Mountain Goat Software.

[Shepperd et al. 2001] Shepperd, Martin, Cartwright, Michelle, *Predicting with Sparse Data*, IEEE Transactions on Software Engineering, VOL. 27, NO. 11, 2001.

[Spencer 1982] Spencer, Ian, *Incomplete Experimental Designs for Multidimensional Scaling*, Chapter 3. In R.G. Golledge and J.N. Rayner (Eds.), *Proximity and Preference: Problems in the Multidimensional Analysis of Large Data Sets*, University of Minnesota Press, 1982.

[Spencer 1983] Spencer, Ian, *Monte Carlo Simulation Studies*, Applied Psychological Measurement Vol. 7, No. 4, 1983.