Behavior Research Methods & Instrumentation
1979, Vol. 11 (2), 147-150

Collecting paired-comparison data with a sorting algorithm

C. P. WHALEY
University of Waterloo, Waterloo, Ontario, Canada

Monte Carlo techniques were used to evaluate the performance of an on-line paired-comparisons data collection procedure that makes use of a common computer sorting algorithm. The results revealed that the sorting method can reduce the number of trials per subject substantially even when a considerable amount of random error is present. While a complete paired-comparisons design requires $N(N-1)/2$ trials (where N is the number of objects), the sorting procedure requires a theoretical minimum of $N(\log_2 N)$ trials. The savings in the number of trials consequently increases with N. Furthermore, the negative effect of random error on the final ordering of the data from the sorting method is small and decreases with the number of stimuli. The data from a small empirical study reinforces the Monte Carlo observations. It is recommended that the sorting method be used in place of the complete paired-comparisons procedure whenever a substantial number of stimuli are included in the design.

In applied psychology and other areas as well, the method of directional paired-comparisons is frequently used to obtain an ordering of a set of N stimuli along some prespecified dimension, commonly preference (although size, weight, utility, or any other scale could be used). The data collection process is generally quite slow since (1) many subjects are usually run to meet the requirements of statistical tests (Maxwell, 1974) or multidimensional unfolding analyses (e.g., Kruskal, Young, & Seery, Note 1), and (2) $N(N-1)/2$ trials are required for each subject to test all possible pairs of N stimuli.

While little can be done about the first problem, a partial solution is offered here for the second. Attaining an ordering of a set of stimuli from an experimental subject is not unlike the problem of ordering a set of scrambled numbers with a computer sorting algorithm. However, while complete paired-comparisons experiments require $N(N-1)/2$ trials, modern sorting algorithms theoretically require as few as $N(\text{Log}_2 N)$ iterations through the critical decision stage where two quantities are compared. If a paired-comparisons experiment could be carried out using a similar algorithm, the savings would be quite substantial, as is shown in Table 1.

What is proposed here is that the experimental subject replace the decision statement in the sorting algorithm. Naturally, such an experiment would have to be run on-line. A typical sorting algorithm requiring little core, and yet fully capable of executing an on-line paired-comparisons experiment, is shown in Table 2. Here pairs of animal names are presented to the subject, who responds "1" or "2" to indicate which member

The author's new address is Behavioural Studies, Department 3Z70, Bell-Northern Research, P. O. Box 3511, Station C, Ottawa K1Y 4H7, Canada.

Table 1
Trials Required for Complete and "Sorted" Paired Comparison Methods

Number of Stimuli	Trials	
	Complete	Sorted
2	1	2
4	6	8
8	28	24
16	120	64
32	496	160
64	2016	384
128	8128	896

Table 2
A BASIC Program for the Execution of an On-Line Paired Comparisons Experiment Using a Sorting Algorithm

```
 10 REM ----- PAIRED COMPARISONS EXPT.
 20 DIM A(10),IX(10)
 30 N=10:FOR I=1 TO N:READ A(I):IX(I)=I:NEXT I
 40 GOSUB 70:PRINT
 50 FOR I=1 TO N:PRINT A(IX(I)):NEXT I:END
 60 REM ------ SORTING ROUTINE
 70 M=N
 80 M=INT(M/2)
 90 IF M=0 THEN RETURN
100 KK=N-M
110 J=1
120 I=J
130 IM=I+M
140 REM ------ STIMULUS PAIRS ARE ASSESSED HERE
150 PRINT:PRINT A(IX(I)),A(IX(IM)):INPUT RS
160 IF RS=1 GOTO 200
170 J=J+1
180 IF J > KK GOTO 80
190 GOTO 120
200 LL=IX(I):IX(I)=IX(IM):IX(IM)=LL:I=I-M
210 IF I < 1 GOTO 170
220 GOTO 130
230 REM ------ STIMULUS LIST
240 DATA ANT,DOG,FROG,TURKEY,SHARK,ZEBRA
250 DATA PENGUIN,BEAR,WHALE,MOOSE
```


0005-7878/79/020147-04$00.65/0

of the pair is greater on the scale specified in the instructions.

This BASIC program, which incorporates the now classic algorithm published by Shell (1959), was implemented and tested on an ISC Intecolor 8051 intelligent terminal with 8K RAM. Less than 750 bytes of memory was required, including the program itself.

What is more critical than being able to set up such an experiment, however, is the effect of human error on the final ordering attained by the program. Since computer sorting algorithms seldom err, the effect that random error might have on the sorting process is unknown. Since the selection of each pair presented is contingent to a large extent on the subject's response to the previous pair, one might expect some errors to be more costly than others. To clarify this situation, a Monte Carlo investigation was carried out.

EXPERIMENT 1

Method

A total of 5,000 experimental sessions were artificially generated. A quasirandom number generator determined when errors occurred. The amount of error was systematically varied: 1%, 5%, 10%, 25%, and 50%. Two limiting cases were considered: (1) where the stimuli (N = 8, 16, 32, 64, or 128) were passed to the algorithm in the original (correct) order, and (2) where the stimuli (with the same five levels of N) were passed to the algorithm in a new random order for each experimental session. A total of 100 experimental sessions were executed for each of the 5 (error levels) X 5 (stimulus sample sizes) X 2 (presentation conditions) = 50 cells.

The dependent variables were tau, the Kendall rank-order correlation between the obtained ordering and the original ordering, and the number of iterations (trials), that is, the number of cycles through the algorithm to obtain a final, although not necessarily correct, ordering.

Results and Discussion

Tables 3 and 4 show the results with mean number of iterations as the dependent variable. Tables 5 and 6 contain mean tau values. The 50% error condition is not presented in any of the tables since the results are of little theoretical interest.[1]

Tables 3 and 4 reveal that the mean number of iterations (trials) increases beyond the theoretical number, $N(Log_2N)$, with both N and the amount of error. It is evident, however, that even in the most extreme cases, a considerable savings is realized, and that when N is less than 32 and the error level is below 10%, the number of trials is less than the theoretical number for the fixed presentation order.

Mean tau across 100 replications, shown in Tables 5 and 6, increases with N for all error levels. This is ideal since the larger the N, the greater the savings in total trials by using the sorting technique.

If we are willing to adopt a mean tau of .90 as a criterion for "good" performance of the algorithm, error rates of up to 10% are allowable regardless of fixed or random presentation for all stimulus set sizes included here.

EXPERIMENT 2

By way of lending empirical support to the findings already presented, a small experiment was carried out

Table 3
Mean Number of Iterations: Fixed Presentation Order

Percent Error	Number of Stimuli									
	8		16		32		64		128	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
1	17.33	.10	50.25	.20	132.39	.35	331.11	.69	795.85	.94
5	18.73	.22	55.26	.48	148.71	.94	375.05	1.56	910.51	2.94
10	20.13	.24	61.17	.61	170.36	1.43	432.11	2.22	1074.06	4.18
25	23.56	.28	73.45	.71	206.11	1.43	549.50	2.28	1372.70	4.29
0*	24.00		64.00		160.00		384.00		896.00	

**Theoretical number of iterations based on N-LOG$_2$-N.*

Table 4
Mean Number of Iterations: Random Presentation Order

Percent Error	Number of Stimuli									
	8		16		32		64		128	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
1	21.65	.22	69.09	.57	195.25	1.49	540.15	3.90	1420.63	10.46
5	22.45	.25	71.28	.62	205.94	1.57	557.83	3.25	1430.18	6.73
10	23.38	.23	74.50	.58	215.03	1.60	563.93	2.73	1465.99	4.83
25	24.11	.30	77.05	.67	219.31	1.37	573.90	2.32	1431.94	4.30
0*	24.00		64.00		160.00		384.00		896.00	

**Theoretical number of iterations based on N-LOG$_2$-N.*

Table 5
Mean Tau: Fixed Presentation Order

Percent Error	Number of Stimuli: 8 Mean	8 SE	16 Mean	16 SE	32 Mean	32 SE	64 Mean	64 SE	128 Mean	128 SE
1	.992	.002	.997	.001	.999	.000	.999	.000	1.000	.000
5	.964	.006	.984	.002	.989	.001	.995	.000	.997	.000
10	.912	.009	.954	.003	.965	.002	.980	.001	.988	.001
25	.647	.021	.719	.011	.739	.010	.748	.006	.771	.005

Table 6
Mean Tau: Random Presentation Order

Percent Error	Number of Stimuli: 8 Mean	8 SE	16 Mean	16 SE	32 Mean	32 SE	64 Mean	64 SE	128 Mean	128 SE
1	.971	.008	.993	.002	.994	.001	.994	.001	.993	.001
5	.939	.010	.954	.005	.965	.003	.969	.002	.972	.002
10	.890	.013	.899	.008	.915	.005	.924	.004	.925	.003
25	.588	.023	.636	.014	.647	.009	.661	.007	.654	.005

in which subjects judged the apparent size (i.e., area) of pairs of rectangles presented on a CRT screen.

Method

Subjects. Twelve adult subjects (aged 25 to 35 years) volunteered to participate without pay.

Procedure. Seventeen different rectangles were generated. Area and shape (i.e., width/height) were varied orthogonally using an algorithm similar to that used by Krantz and Tversky (1975) for preparing their stimuli. The method relies on using equal-interval logarithmic scale values for width and height.

The experiment consisted of two phases: one wherein a complete $N(N-1)/2$ paired-comparison procedure was carried out, and one where the sorting algorithm was employed. All subjects participated in both phases.

An ISC Intecolor 8051 with 8K RAM was programmed in BASIC to conduct the experiment. The built-in graphics routines were used to construct the rectangles. One rectangle of each pair was presented in green, the other in yellow. The rectangles appeared in randomly selected nonoverlapping locations on the 19-in. (diagonal measurement) screen. The viewing distance was approximately 30 in. The subject chose the larger member of each pair and responded by pressing a key of the same color as the rectangle chosen. The next pair of rectangles was presented as soon as a response was detected by the machine.

While no objective time limits were enforced, subjects were encouraged to respond to each pair in less than 5 sec. Informal observation by the experimenter suggested that this guideline was only rarely violated. At this rate of response, the complete experimental session for each subject lasted approximately 15 min.

The order of presentation of pairs for both phases of the experiment was randomized differently for each subject prior to beginning. The choice of color to represent the members of each pair was also randomly determined. One half of the subjects (randomly assigned) participated in the full paired-comparison phase first and the "sorted" paired-comparison phase second. The remainder participated with the phase order reversed. There was a brief 15- to 30-sec pause between phases as the computer reinitialized a large number of parameters. From the subject's perspective, the two phases appeared the same, although a few subjects commented that one half was noticeably shorter than the other.

Results

Table 7 contains descriptive statistics based on the combined data of the 12 subjects who participated. As anticipated, the mean number of trials per subject was substantially lower with the sorting method than with the complete paired-comparisons procedure.

In terms of performance relative to the actual area of the rectangles, however, both proportion correct and mean tau were significantly higher for the complete method than for the sorting method [$t(11) = 10.90$, $p < .01$, and $t(11) = 3.03$, $p < .01$, respectively].

While these findings tend to diminish to some extent the results presented in Experiment 1, there are several factors that help to clarify the situation. First, the proportion correct can be expected to be lower for the sorting method than for the complete method. Because of the nature of the algorithm itself, the pairs of stimuli that are selected for presentation from the complete set tend to rapidly converge on those that are minimally

Table 7
Descriptive Statistics for Rectangle Size Experiment

	Paired Comparison Method: Sorting Mean	Sorting SD	Complete Mean	Complete SD
Number of Trials	83.33	5.31	136.00	.00
P(Correct)	.75	.04	.86	.04
Correlation with Area*	.77	.09	.82	.06

**Kendall's tau.*

different. Consequently, the proportion of errors would be expected to be higher due to task difficulty. Second, while the difference is nonetheless significant, the mean tau values are considerably closer in magnitude than are the proportion correct figures. Moreover, the mean tau value for the sorting method (.77) is considerably greater than the .64 one might be led to expect from the most comparable cell in the Monte Carlo data (Table 6). Finally, the difference in subjects' performance for the two tasks can be shown to be virtually nonexistent when the two sets of ranks for all subjects are scaled prior to statistical evaluation.

The ranks for the rectangles obtained from the respective methods were converted to interval scale values using a method described by Maxwell (1974) and incorporated in a computer program package called PCSTAT (Whaley, 1977). When the interval scaled values for the two methods are then correlated, the differences noted earlier effectively disappear. The Pearson r between methods was .983. Furthermore, Pearson r = .968 between objectively measured area and the scaled values from the complete method, and r = .976 between area and the scaled sorting values.

This means that in using the sorting method with subsequent scaling, 95% of the variance in the rectangle area was captured from the data of just 12 subjects who failed to respond correctly 25% of the time.

GENERAL DISCUSSION

The performance of the sorting procedure relative to the traditional method is generally very good, and, as has been shown with the Monte Carlo data, increases with the number of stimuli included in the design.

It appears to be the case from the rectangle size experiment, however, that even with fairly small stimulus sets good performance can be expected with the subsequent application of scaling procedures. This implies that human errors in experiments of this kind are associated with small differences in the stimuli and are, as such, not random, unlike those in the Monte Carlo experiment. Thus, one can expect at very least equal and most likely better performance than that indicated in Tables 3-6.

REFERENCE NOTE

1. Kruskal, J. B., Young, F. W., & Seery, J. B. *How to use KYST, a very flexible program to do multidimensional scaling and unfolding.* Unpublished manuscript. Bell Laboratories, Murray Hill, New Jersey, 1973.

REFERENCES

KRANTZ, D. H., & TVERSKY, A. Similarity of rectangles: An analysis of subjective dimensions. *Journal of Mathematical Psychology*, 1975, **12**, 4-34.

MAXWELL, A. W. The logistic transformation in the analysis of paired comparison data. *British Journal of Mathematical and Statistical Psychology*, 1974, **27**, 62-71.

SHELL, D. L. A high-speed sorting procedure. *Communications of the ACM*, July 1959, 30-32.

WHALEY, C. P. PCSTAT: Statistical analysis of paired-comparison data. *Behavior Research Methods & Instrumentation*, 1977, **9**, 372.

NOTE

1. As one would expect, the mean tau values were all approximately zero for this case. The 50% error condition ought to require more trials than any other, and thereby indicate the length of an experiment in the worst possible set of conditions. In fact, this was the case, but there was no appreciable difference between the 50% error condition and the 25% error condition in the number of iterations required.