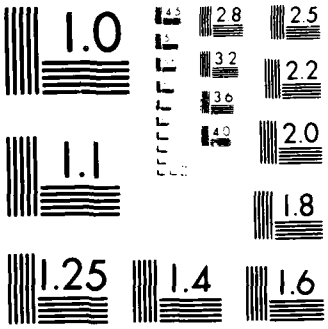
AD-A186 815 PRACTICAL APPLICABILITY OF EXACT AND APPROXIMATE FORMS OF THE RANDOMIZATION TEST FOR TWO INDEPENDENT SAMPLES (U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA D H HESSE 1/1

UNCLASSIFIED SEP 87 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART

DTIC FILE COPY

(2)

AD-A186 815

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DTIC
ELECTE
NOV 3 0 1987
S D

THESIS

PRACTICAL APPLICABILITY OF EXACT AND APPROXIMATE FORMS OF THE RANDOMIZATION TEST FOR TWO INDEPENDENT SAMPLES

by

Derek H. Hesse

September 1987

Thesis Coadvisors: F. R. Richards
H. M. Fredricksen

Approved for public release; distribution is unlimited

87 11 2 289

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) Practical Applicability of Exact and Approximate Forms of the Randomization Test for Two Independent Samples

12 PERSONAL AUTHOR(S) HESSE, Derek H.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1987 September	58

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GROUP	Statistics, Randomization Tests, Computational Efficiency, #P-Complete Problems

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The practical applicability of randomization tests is discussed. The randomization test for two independent samples is the specific test examined in both hypothesis and significance testing contexts. This test has optimum theoretical properties as a nonparametric procedure for comparing the means of two populations. However, the calculations that are required to actually use the test in practice can be extremely time consuming. Using the randomization test for two independent samples to conduct a significance test is shown to be a #P-complete enumeration problem. This implies that a computationally efficient way to perform an exact version of the procedure is not likely to exist. Two approximate ways to perform the randomization test are studied with the aid of a simulation. One method uses a normal distribution to

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. F. R. Richards	22b TELEPHONE (Include Area Code) 408-372-3439 22c OFFICE SYMBOL 55Rh

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Block 19 Abstract Continued

approximate the actual randomization distribution and the other method is the usual two sample t-test. The t-test is found to yield results very close to those that are obtained from the exact randomization test under the conditions studied.

A-1

Approved for public release; distribution is unlimited.

Practical Applicability of Exact and Approximate Forms of the Randomization Test for Two Independent Samples

by

Derek H. Hesse
Lieutenant, United States Navy
B.S., U.S. Naval Academy, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

Author: _____
Derek H. Hesse

Approved by: _____
F.R. Richards, Co-Advisor

_____
H.M. Fredricksen, Co-Advisor

_____
Peter Purdue, Chairman,
Department of Operations Research

_____
James M. Fremgen,
Acting Dean of Information and Policy Sciences

ABSTRACT

The practical applicability of randomization tests is discussed. The *randomization test for two independent samples* is the specific test examined in both hypothesis and significance testing contexts. This test has optimum theoretical properties as a nonparametric procedure for comparing the means of two populations. However, the calculations that are required to actually use the test in practice can be extremely time consuming. Using the randomization test for two independent samples to conduct a significance test is shown to be a **#P-complete** enumeration problem. This implies that a computationally efficient way to perform an exact version of the procedure is not likely to exist. Two approximate ways to perform the randomization test are studied with the aid of a simulation. One method uses a normal distribution to approximate the actual randomization distribution and the other method is the usual two sample t-test. The t-test is found to yield results very close to those that are obtained from the exact randomization test under the conditions studied.

TABLE OF CONTENTS

I. INTRODUCTION 9

II. RANDOMIZATION TEST THEORY 11

A. THE RANDOMIZATION TEST FOR TWO INDEPENDENT SAMPLES 11

1. Randomization Concept 11

2. Test Method 11

B. THEORETICAL PROPERTIES 12

1. Efficiency and Asymptotic Relative Efficiency 12

2. Unbiasedness 14

3. Uniformly Most Powerful Test 14

III. COMPUTATIONAL ISSUES 16

A. COMBINATORIAL NATURE OF THE RANDOMIZATION TEST 16

1. Rapid Growth of Combinations 16

2. Computer Time Considerations 17

B. ALGORITHMIC EFFICIENCY AND NP-COMPLETENESS 18

1. Basic Concepts and Terminology 18

2. Polynomial Time and Exponential Time Algorithms 20

3. The Classes P and NP 20

4. NP Complete Problems 22

5. #P-Complete Problems 24

C. EFFICIENCY OF ALGORITHMS FOR PERFORMING THE RANDOMIZATION TEST 25

1. Randomization Test is an Enumeration problem 25

2. Significance Testing is #P-Complete 26

IV. ANALYSIS OF AN APPROXIMATE RANDOMIZATION TEST PROCEDURE 29

A. INTRODUCTION 29
1. Reasons For Using An Approximate Method 29
2. Considerations When Using Approximations 29
3. Method Studied 30
B. PERFORMING APPROXIMATE RANDOMIZATION TESTS 30
1. Subsampling 30
2. Blocks 31
3. T-test as an Approximation 31
4. 2-Moment Fit Method 32
C. A COMPARISON OF EXACT AND APPROXIMATE METHODS USING SIMULATION 34
1. Purpose of Simulation 34
2. Description of Simulation 35
3. Results and Interpretation 36
4. Summary of Results 39

V. SUMMARY 42
A. MAJOR RESULTS AND CONCLUSIONS 42
B. AREAS FOR FURTHER RESEARCH 44
1. Approximations 44
2. Pseudo-Polynomial Time Algorithms 44

APPENDIX A: DERIVATION OF THE 2-MOMENT FIT METHOD 46

APPENDIX B: SIMULATION PROGRAM LISTING 50

LIST OF REFERENCES 56

INITIAL DISTRIBUTION LIST 57

LIST OF TABLES

1. COMBINATIONS 16
2. COMPUTATION TIMES 17

LIST OF FIGURES

3.1 Relationships Between Classes of Problems 23
4.1 Typical Randomization Histogram 32
4.2 Normal Density Fitted to Randomization Histogram 33
4.3 Tail Areas Correspond to α 33
4.4 Significance Level Comparisons: H_0 True 37
4.5 Significance Level Comparisons: H_0 False 38
4.6 Example Power Curves For $\alpha_0 = .05$ 40
4.7 Multimodal Histograms 41

I. INTRODUCTION

Randomization tests have long been recognized as powerful nonparametric statistical methods since the introduction of the principal ideas by R.A. Fisher in 1935. Even when compared to the most powerful parametric tests such as the t-test, randomization tests perform extremely well. Theoretical work since Fisher's paper has indicated that randomization tests may be the best methods to use in many situations involving significance testing or tests of hypotheses. This is particularly true if assumptions about the underlying probability distributions are difficult to establish.

Despite their strong theoretical basis, however, randomization tests have not been in widespread use. The major reason they have not been commonly used is because they are very tedious to perform. Even when sample sizes are relatively small, the computation time required to perform these tests can be significant. While this is less of a problem with modern computing equipment, there still exists a point where the size of the data sets is large enough to make the procedures impractical. This point is reached rapidly due to the inherent combinatorial nature of the algorithms used to perform the tests. Vast improvements in computational speed have only a marginal effect on the size of the data sets that can be handled. Approximate randomization tests have been developed because of these difficulties, but analytic results describing the errors involved with their use are limited. Exact analytic results are difficult to obtain because the form of the underlying distributions is not known.

This thesis addresses the issue of practical implementation of randomization tests. The *randomization test for two independent samples* is the specific procedure chosen for the entire study. This procedure is representative of randomization tests in general. A complete description of this test, along with each assumption needed to ensure its validity is given first. Also included is a summary of some of the important theoretical work that has been done since the test appeared in the literature. Next, the methods available for performing an exact version of this test are shown to require so much computation time when the length of the input data sets increases that the methods become impractical on even the fastest computers. The mathematical framework necessary to prove this result is fully developed using concepts from the theory of **#P**-complete enumeration problems.

Finally, an approximate method for performing the randomization test for two independent samples is described. This method is compared to the exact test and the standard t-test, using the same sample values for each. The samples are generated from several distributions through standard simulation routines and the performance of each test in terms of significance level and average power is recorded. The results from this simulation are discussed, and recommendations are made as to which test should be used and when.

II. RANDOMIZATION TEST THEORY

A. THE RANDOMIZATION TEST FOR TWO INDEPENDENT SAMPLES

1. Randomization Concept

The basic idea of *randomization* was introduced by Fisher in 1935 [Ref. 1]. Randomization involves taking precautions in the design and actual performance of an experiment to ensure the validity of statistical procedures used on the resulting data. A *randomized experiment* is one in which treatments are randomly assigned within each block. Fisher argued that, on the basis of a randomized experiment, it is possible to conduct a test of significance without making any assumptions about the distribution (a distribution free procedure) [Ref. 2: p. 95]. The idea of using a *randomization test* is to perform a hypothesis or significance test involving two or more samples from populations whose distribution functions are unknown. The hypotheses of interest usually take the form of testing whether or not these distribution functions are all identical, except for possibly different location parameters (means, for example).

2. Test Method

A *randomization test for two independent samples* was first proposed by Pitman [Ref. 3]. The purpose of this test is to compare the means of two populations. The procedure is to draw two random samples $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ and $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m$ of sizes n and m respectively from two independent populations $\mathbf{X}$ and $\mathbf{Y}$. Following the description in Conover [Ref. 4: p. 328], independence *within* each sample is assumed, as well as independence *between* the two samples. Also assumed is that either the two population distribution functions are identical, or one population has a larger mean than the other. Without this second assumption the test is still valid but might lack consistency. The hypothesis to be tested is that the mean of the population from which the $\mathbf{X}$'s were drawn ($\boldsymbol{\mu}_x$) is the same as the mean of the population from which the $\mathbf{Y}$'s were drawn ($\boldsymbol{\mu}_y$). The (two-tailed) alternative is that the means are not the same. In other words, this is equivalent to testing

$$H_0: \boldsymbol{\mu}_x = \boldsymbol{\mu}_y$$
$$\text{vs. } H_1: \boldsymbol{\mu}_x \neq \boldsymbol{\mu}_y$$

where H_0 denotes the null hypothesis and H_1 is the alternate. This two-tailed test is the specific form of the randomization test that will be referred to henceforth.

An appropriate test statistic that can be used is just the sum of the **X** observations:

$$T_0 = \sum_{i=1}^{n} X_i \qquad \text{(eqn 2.1)}$$

The *critical* (or *significance*) level of the test is denoted α. This number is equal to the probability that the test statistic could have produced values identical to or more extreme than the originally observed value T_0. To find α, the null hypothesis H_0 is assumed to be true; that is, the **X** and **Y** populations are identically distributed. If H_0 is true, then the **X**'s should have no more of a tendency to be low or high than do the **Y**'s. Essentially, the **X**'s and **Y**'s could be thought of as just one collection of $n+m$ observations from the same distribution, and each selection of n **X** observations should be considered equally likely from the $n+m$ observations available.

The significance level α is obtained by counting the number of ways n of the $n+m$ observations may be selected so that their sum is equal to or more extreme than the originally observed value of the test statistic T_0. *More extreme* means smaller if T_0 is in the lower tail or larger if T_0 is in the upper tail of the distribution of all possible values of the test statistic using the observed data. The number of ways is doubled, because the test is two-tailed, and divided by $\binom{n+m}{n}$ to yield α. [Ref. 4: p. 329]

In the case of *hypothesis testing*, a critical value, say α_0, is specified beforehand and the null hypothesis is rejected if $\alpha < \alpha_0$. If *significance testing* is being performed, the interpretation is somewhat different. In this case, there is no pre-specified value α_0. The significance level α is computed and if it is small, say less than .01, then either the observed value of the test statistic happened to be a rare event *or* the basic premise that the **X**'s and **Y**'s are identically distributed is unlikely. The smaller α is, the more compelling is the latter event.

B. THEORETICAL PROPERTIES

1. Efficiency and Asymptotic Relative Efficiency

The term *efficiency* is applied to statistical tests when comparing the sample sizes required by two different tests that give comparable results. The *power* of a test is defined to be the probability of rejecting the null hypothesis H_0 when it is false. The

power of a test depends upon factors such as sample size and the particular alternate hypothesis H_1 chosen. Suppose two tests have the same level of significance and power and they can both be used to test a particular H_0 against a particular alternate H_1. Then the test requiring the smaller sample size is preferred, because a smaller sample size means less cost and effort is required in the experiment. As indicated in Conover [Ref. 4: p. 88], the test with the smaller sample size is said to be *more efficient* than the other test.

Suppose T_1 and T_2 represent two tests that could be used to test a given H_0 against a given H_1. Suppose further that either test, if used, would yield the same value of α and the same power characteristics. Then, adopting Conover's notation [Ref. 4: pp. 88-89], the *relative efficiency* of T_1 to T_2 is the ratio n_2/n_1, where n_1 and n_2 are the sample sizes required by the tests T_1 and T_2 respectively in order for each to yield identical results.

The relative efficiency of two tests depends on the particular values chosen for α and power and it also depends on the particular alternate hypothesis H_1 chosen if H_1 is composite. A *composite* hypothesis is one that does not specify a probability law completely. It would be more useful if an efficiency measure could be developed that does not depend on these quantities. Such a measure can be developed in the following way. Consider two parallel *sequences* of tests constructed so that as n_1 and n_2 are increased, the significance level and power of each pair of tests remains the same. To accomplish this, two things would be required. First, as n_1 is increased, the power of each test in the first sequence would change if the alternate hypothesis H_1 were kept fixed. To keep the power constant, a different H_1 could be selected each time. The values of α and power would then remain the same from test to test in the first sequence. Second, for each value of n_1, a value of n_2 must be calculated so that each test in the second sequence has the same values of α and power as its corresponding test in the first sequence under the alternative hypothesis chosen. Then there is a sequence of values of relative efficiency n_2/n_1, one for each pair of tests in the original sequences. If n_2/n_1 approaches a constant as n_1 becomes large, then that constant is called the *asymptotic relative efficiency* (A.R.E.) of the first sequence of tests to the second, if the constant is the same for all values of α and power.

The A.R.E. is one measure of a test's performance. For many nonparametric tests, the A.R.E. is less than 1.0 when compared to the corresponding parametric tests in situations where they are appropriate. This implies that, in general, a nonparametric

procedure will require a larger sample size to achieve the same results as a parametric procedure *if the basic assumptions of the parametric method are valid* (e.g., normality). However, according to Conover [Ref. 4: p. 327], the A.R.E. of the randomization test is 1.0 when compared to the most powerful parametric tests in some situations. The A.R.E. may be much higher than 1.0 if the basic assumptions of the parametric test are not met. Thus a randomization test should be at least as efficient as a parametric test and could be more efficient on the basis of asymptotic relative efficiency. Note that, on the basis of *relative efficiency* (not asymptotic), the randomization test might be better or worse than a parametric test depending on the circumstances. Generally, though, asymptotic relative efficiency is a reasonable and widely accepted measure of a test's performance.

2. **Unbiasedness**

The definition of an *unbiased test* is a test in which the probability of rejecting a false H_0 is always greater than or equal to the probability of rejecting a true H_0 [Ref. 4: p. 86]. Another way to state this is to say the power is at least as large as the level of significance. This is obviously a desirable property to have; a test should be more likely to reject H_0 when it is false than when it is true. The randomization test has been shown to be an unbiased test in Lehmann and other sources [Refs. 5,6].

3. **Uniformly Most Powerful Test**

The *power* of a test, denoted by $1-\beta$, is the probability of rejecting a false null hypothesis. In the case of a simple alternate hypothesis that specifies a probability law completely, this is a unique number. However, in the case of a composite alternate hypothesis, the power is not unique. The alternate hypothesis being considered here, H_1: $\mu_x \neq \mu_y$, is composite since there are an infinite number of possible probability functions implied by the inequality. When the alternate hypothesis is of composite type, power is represented by a *power function*, where the value of power depends on the parameters of the alternate probability laws implied by H_1. Specifically,

$$\text{Power} = P(\text{Reject } H_0 | \theta) \qquad \text{(eqn 2.2)}$$

Where $\theta = \mu_x - \mu_y$. The power function for a two-tailed test of H_0 vs. H_1 has a characteristic 'U'-shape centered at the value $\mu_x - \mu_y = 0$.

The *size* of a test is defined to be the maximum probability of a Type **I** error (rejecting the null hypothesis H_0 when it is true) over all values of parameters for

which H_0 is true. Among *all* tests which have size α, the best test (if it exists) is that test which has the largest power over all values for which H_1 is true. Such a test is called the *uniformly most powerful* test of size α [Ref. 7]. Graphically, this implies the power function of a uniformly most powerful test will pass through the point α when H_0 is true and will lie above the power curves of all other possible tests of size α that could be used.

In the case of the randomization test for two independent samples, Oden and Wedel [Ref. 5: p. 520] have stated the following for the case of a one-sided alternative H_1: "Among all unbiased tests for testing H_0 against H_1 the test is uniformly most powerful for the subclass of H_1 with elements (f, g) such that ln (fg) is linear, including e.g. the case of 'normality and equal variances'." The extension to a two-sided alternative is readily apparent. Here, f and g are one-dimensional probability density functions that belong to the class of functions associated with H_1. An example of densities f and g that satisfy such conditions would be two standard exponential density functions with parameters λ_1 and λ_2, respectively.

This is a very significant result. The fact that the randomization test for two independent samples is the uniformly most powerful test against a certain subclass of alternatives is strong theoretical justification for use of the test in many circumstances. When the other desirable properties of the test mentioned previously are also considered, the implication is that the randomization test should be preferred over any other method of comparing means unless underlying distributions can be clearly justified.

III. COMPUTATIONAL ISSUES

A. COMBINATORIAL NATURE OF THE RANDOMIZATION TEST

1. Rapid Growth of Combinations

Even though the randomization test for two independent samples has been shown to possess many desirable properties, the test is not encountered very often in practice. The basic reason is the amount of computation time required to perform the test. In the previous chapter, the test method was shown to be essentially a counting procedure involving combinations of the data. The number of combinations possible of $n+m$ objects taken n at a time is $\binom{n+m}{n}$, and this number grows at a substantial rate as n and m are increased. The following table illustrates the growth of combinations for some selected values of n and m:

TABLE 1
COMBINATIONS

n	m	$\binom{n+m}{n}$
2	2	6
5	5	252
7	8	6435
9	10	92378
11	12	1352078
15	20	3247943160

There is no known way to perform the exact randomization test for the general case other than enumerating all possible combinations of the data (or at least a fair proportion of them) and comparing each one to the original test statistic T_0. In certain special cases, more efficient methods do exist. For an example of such a method see Soms [Ref. 8]. It is possible to reduce the number of combinations that need to be considered through the use of more intelligent enumeration schemes, backtrack search or other techniques. However, even though considerable savings could be achieved, the number of combinations remaining continues to grow at a rate proportional to total enumeration. Thus the computation time required to perform the general randomization test is a function of the number of combinations involved.

2. Computer Time Considerations

As an example of how rapidly enumeration becomes untenable, consider a computing device capable of generating combinations of data sequentially and comparing each one to a fixed value. Assume that each combination could be formed, compared, and counted in a time span of 1 microsecond (this is very fast, even for a large computer). Also assume it is desired to use this device to perform the randomization test on samples of sizes up to $n = 30$ and $m = 30$. Such sample sizes are very common in practice. The following table gives the total time that would be required to enumerate all combinations of the form $\binom{n+m}{n}$ using this device. For simplicity, only equal sample sizes are included ($n = m$):

TABLE 2
COMPUTATION TIMES

n or m	Approximate Time Requirement
5	.00025 seconds
10	.18 seconds
15	155 seconds
20	38.3 hours
25	4.01 years
30	37.5 centuries

Even if the number of combinations could be reduced by a factor of 100 through careful enumeration or backtrack search as mentioned before, the time requirements would remain virtually untenable. Further, if a new computing device were installed that performed the calculations 1000 times faster, our ability to process the data sets would be increased only marginally.

The examples above demonstrate that the direct method of performing the randomization test for two independent samples is not efficient in any reasonable sense of the word. As sample sizes increase, the inefficiency of the method makes it unsuitable for practical use. In the next section, it is shown that *no* efficient algorithm is likely to exist for performing this test. To define what is meant by an *efficient algorithm*, some ideas from the theory of **NP**-completeness are introduced.

B. ALGORITHMIC EFFICIENCY AND NP-COMPLETENESS

1. Basic Concepts and Terminology

To begin a discussion of algorithmic efficiency, several basic terms must be defined. An excellent treatment of the subject is given in Garey and Johnson [Ref. 9], and the terminology used there is adopted herein. Following Garey and Johnson, an *algorithm* is a step-by-step procedure used to solve a problem. A *problem* is

> . . . a general question to be answered, usually possessing several *parameters*, or free variables, whose values are left unspecified. A problem is described by giving: (1) a general description of all its parameters, and (2) a statement of what properties the answer, or *solution*, is required to satisfy. An *instance* of a problem is obtained by specifying particular values for all the problem parameters. . . . An algorithm is said to *solve* a problem Π if that algorithm can be applied to any instance I of Π and is guaranteed always to produce a solution for that instance I.

To show the use of the above terminology, consider two classic problems from graph theory. The first is due to the 19th century mathematician William Rowan Hamilton. The problem is to decide if an arbitrary graph consisting of a collection of vertices and edges has a path that passes through each vertex exactly once. Such a path, if it exists, is known as a *Hamiltonian path*. The parameters of this problem consist of a finite set $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ of vertices and a set $E = \{e_1, e_2, \ldots, e_j\}$ of edges between pairs of vertices. A solution is an ordering $< \mathbf{v}_{(1)}, \mathbf{v}_{(2)}, \ldots, \mathbf{v}_{(k)} >$ of the vertices such that $(\mathbf{v}_{(i)}, \mathbf{v}_{(i+1)}) \in E$ for $1 \leq i < k$ and each vertex is visited exactly once. An instance of the problem would be obtained by giving specific vertices and edges (referenced to a coordinate system, for example).

The second problem, due to Euler, is very similar to Hamilton's problem. It can be stated using the same sets as above, except that in this case, a path is sought which traverses each *edge* in the graph exactly once. Such a path is called an *Eulerian path*. Both Hamilton's problem and Euler's problem can be solved by exhaustive tabulation of all possible paths, checking each one to see if it has the required properties. This approach has the same problems as complete enumeration of combinations in the randomization test. The number of possible paths grows in a similar fashion, and the algorithm quickly becomes too inefficient for practical use.

The important distinction between these two graph theoretic problems is that there is a *much* easier way to solve Euler's problem than exhaustive tabulation. Euler showed that a path traversing each edge of a graph exactly once must exist if the graph

meets two conditions: (1) the graph must be connected and (2) there must be an even number of edges that meet at any vertex, with the exception of the starting and finishing points of the path. The computation time required to check this is related to the number of vertices and edges, not the number of possible paths. An algorithm using this approach is practical even when the number of vertices and edges is very large, despite the fact that the number of possible paths may be astronomical. In the case of Hamilton's problem, however, no such simple and efficient method of solution has ever been found. As discussed by Lewis and Papadimitriou [Ref. 10: p. 102], the most efficient methods available today are fundamentally no better than exhaustive tabulation.

An algorithm that operates 'efficiently' could be viewed as one that uses a minimum amount of computer resources to arrive at the solution to a problem. Computer resources include things such as memory space, CPU time, and I/O (Input/Output) capacity. However, since the critical resource is usually time, the 'most efficient' algorithm is normally the fastest one. The time requirements of an algorithm can be expressed in terms of a single variable, the 'size' of a problem instance. Informally, this can be thought of as the amount of data that must be input to describe a given instance. Examples would be the number of vertices and edges in Hamilton's problem or the number of **X** and **Y** observations in the randomization test. The formal way to characterize problem size views the situation from the standpoint of actual entry into a computing device. Problems must be input in a single finite string of symbols chosen from a fixed set, or *input alphabet*. An *encoding scheme* must be specified, which maps problem instances into the symbolic strings describing them. The *input length* for an instance of a problem is the number of symbols required to specify the instance under the given encoding scheme. As indicated in Garey and Johnson [Ref. 9: pp.5-6], the input length is what is used as the formal measure of instance size.

The *time complexity function* for an algorithm expresses its time requirements by giving, for each possible input length, the largest amount of time needed by the algorithm to solve a problem instance of that size. This function won't be well defined unless a particular computing device, input alphabet and encoding scheme are specified. However, it turns out that these are relatively unimportant factors. What *is* important is the form of the time complexity function. The following discussion from Garey and Johnson [Ref. 9: p.6] introduces this idea:

Different algorithms possess a wide variety of different time complexity functions, and the characterization of which of these are 'efficient enough' and which are 'too inefficient' will always depend on the situation at hand. However, computer scientists recognize a simple distinction that offers considerable insight into these matters. This is the distinction between polynomial time algorithms and exponential time algorithms.

2. Polynomial Time and Exponential Time Algorithms

A *polynomial time algorithm* is defined to be one whose time complexity function can be bounded by a polynomial. That is, there exists a constant c such that

$$|f(N)| \leq c\,|p(N)| \qquad \text{(eqn 3.1)}$$

for all values of $N \geq 0$, where f(N) is the time complexity function, p(N) is a polynomial function of N, and N is the input length. An algorithm whose time complexity function cannot be so bounded by any finite degree polynomial is called an *exponential time algorithm* [Ref. 9: p.6].

The distinction between these two types of algorithms becomes important when the input lengths become large. Polynomial functions of degree k will evaluate to be of the order N^k, but exponential functions are allowed to have terms such as 2^N or $N!$. There is always a value of N beyond which exponential functions grow at a faster rate than any polynomial function, even if the polynomial is of degree 100. It is for this reason that polynomial time algorithms are generally regarded as being much more desirable than exponential time algorithms. There are some notable exceptions, however. As mentioned in Garey and Johnson [Ref. 9: p.9], the simplex algorithm for linear programming has been shown to have exponential time complexity, but it typically runs very quickly in practice. Garey and Johnson [Ref. 9: p.8] also observe that "time complexity as defined is a *worst case* measure, and the fact that an algorithm has time complexity 2^n means only that at least one problem instance of size n requires that much time." Examples of exponential time algorithms that run well in practice are rare. Most exponential time algorithms are variations on exhaustive search or complete enumeration, while polynomial time algorithms generally exploit some fundamental structure of a problem.

3. The Classes P and NP

Problems for which only exponential time algorithms exist are *intractable*, in a sense, because even fairly small instances may never be solved in a realistic amount of

time. For those problems that have polynomial time algorithms, the polynomials involved typically are not of a high order, and thus instances of practically any size can be solved. It would be convenient if all problems of interest could be placed into two groups, those having exponential time complexity and those having polynomial time complexity. Unfortunately, it is exceedingly difficult to *prove* that a given problem is intractable; that is, no polynomial time algorithm can ever be devised to solve it. For a small number of problems it has been shown that exponential time algorithms are the only ones possible, but for most practical problems of interest, this has not been done.

Those problems for which polynomial time algorithms are known to exist are in a class denoted **P**. Euler's problem is a member of **P**. In between this class and the class of provably intractable problems is another class, denoted **NP**. Formal definitions of these classes usually involve models of computation known as Turing machines. However, to gain an understanding of the class **NP**, the concepts of nondeterministic computation and polynomial time verifiability are most important.

A *deterministic* algorithm can be thought of as being composed of a predetermined sequence of operations that do not vary each time the algorithm is used. A *nondeterministic* algorithm introduces the possibility of randomness at points within the procedure. A convenient way to view the operation of such an algorithm is to think of it as being composed of two separate stages, the first being a *guessing stage* and the second a *checking stage*. Given a problem instance, the first stage guesses some structure. The second stage checks this structure in a deterministic fashion to see if it is a solution to the problem. A nondeterministic algorithm is said to operate in polynomial time if there exists some guessed structure that solves the problem and this structure can be verified by the checking stage in polynomial time [Ref. 9: pp.28-29].

The class **NP** is defined informally to be the class of all decision problems that can be 'solved' by polynomial time nondeterministic algorithms [Ref. 9: p.29]. A decision problem is one that has only a *yes* or *no* answer; for example, "Does this graph have a Hamiltonian path?". Most problems of interest can be carefully phrased as decision problems, so this is not overly restrictive. A nondeterministic algorithm would 'solve' Hamilton's problem in the following way: (1) an arbitrary path through the graph would be guessed, and (2) the path would be examined to see if it passes through each vertex exactly once. If the graph does have a Hamiltonian path, then one of the guesses will lead the algorithm to respond 'yes', thus solving the problem. Hamilton's problem is known to be a member of the class **NP**; this implies that step (2) above can be performed in polynomial time.

It is very important to note that the word 'solve' as used above does *not* mean that a nondeterministic algorithm is a realistic method for solving decision problems. This is a only theoretical concept. In fact, a hypothetical machine using a nondeterministic algorithm is envisioned as having the ability to pursue an *unbounded* number of independent computational sequences in parallel. Thus, in Hamilton's problem, the fact that there may be an exponential number of possible paths to check is not counted. It is only required that, given a path, it can be checked in polynomial time. It is this notion of polynomial time *verifiability* that the class **NP** is intended to capture. Most importantly, as Garey and Johnson [Ref. 9: p.12, pp.28-29] point out, polynomial time verifiability does not imply polynomial time solvability.

4. NP Complete Problems

A simplistic way to view the class **NP** is to think of it as containing 'hard' problems: those for which polynomial time algorithms are not known, but neither can it be proved that none exist. The problems in this class also share the important property that any one solution arrived at by 'guessing' can be quickly checked, even though there may be exponentially many guesses possible. The class **P** contains 'easy' problems in the sense that polynomial time algorithms are known for them.

The relationship between **P** and **NP** is fundamental to discussions of algorithmic efficiency. It can easily be shown that $\mathbf{P} \subseteq \mathbf{NP}$. Following Garey and Johnson [Ref. 9: p.32]:

> Every decision problem solvable by a polynomial time deterministic algorithm is also solvable by a polynomial time nondeterministic algorithm. To see this, one simply needs to observe that any deterministic algorithm can be used as the checking stage of a nondeterministic algorithm. If $\Pi \in \mathbf{P}$, and A is any polynomial time deterministic algorithm for Π, we can obtain a polynomial time nondeterministic algorithm for Π merely by using A as the checking stage and ignoring the guess. Thus $\Pi \in \mathbf{P}$ implies $\Pi \in \mathbf{NP}$.

It is widely believed that the inclusion is proper, that is, $\mathbf{P} \subseteq \mathbf{NP}$ but $\mathbf{P} \neq \mathbf{NP}$. This has not been proven, but all evidence seems to strongly suggest this is the case. This is of prime importance, because if **P** differs from **NP**, then the set $\mathbf{NP} - \mathbf{P}$ would not be empty - it would contain intractable problems.

Another concept central to the discussion of algorithmic efficiency is that of problems of *equivalent difficulty*. If several problems can be shown to be related, or of equivalent difficulty, then results of considerable generality and power can be obtained. Referring again to Garey and Johnson [Ref. 9: p.13]:

> The principal technique used for demonstrating that two problems are related is that of 'reducing' one to the other, by giving a constructive transformation that maps any instance of the first problem into an equivalent instance of the second. Such a transformation provides the means for converting any algorithm that solves the second problem into a corresponding algorithm for solving the first problem.

The important characterization here is *polynomial time reducibility*, that is, reductions for which the required transformation can be executed by a polynomial time algorithm. If one problem can be reduced to another through a polynomial time reduction, this ensures that any polynomial time algorithm for the second problem can be converted into a corresponding polynomial time algorithm for the first problem.

There is a subclass of problems within **NP** that has an important property: every problem in **NP** can be polynomially reduced to one of the problems in this subclass. The problems in this subclass are named ***NP-complete*** problems. The implications of this subclass are far-reaching. If any one of the **NP**-complete problems can be solved with a polynomial time algorithm, then so can every problem in **NP**. Also, if any problem in **NP** is intractable, then all the **NP**-complete problems must be intractable. In a sense, the **NP**-complete problems are the 'hardest' problems in **NP**. A picture representing the relationships between the classes of problems discussed so far is given in Figure 3.1.

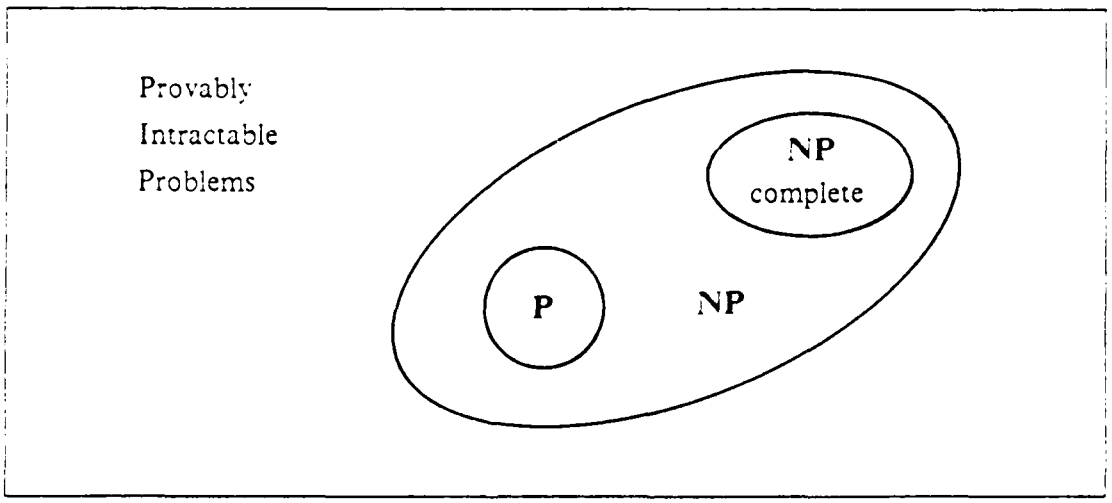


Figure 3.1 Relationships Between Classes of Problems.

Hundreds of problems have been shown to be **NP**-complete since the first such problem was identified by Stephen Cook in 1971 (the 'satisfiability' problem of Boolean logic) [Ref. 9: p.13, p.38]. The list of **NP**-complete problems includes Hamilton's problem, many well known combinatorial problems, and others from a wide variety of disciplines. As more and more problems are added to the list, it appears more and more likely that **P**≠**NP** and the **NP**-complete problems are truly intractable, but little progress has been made toward either a proof or a disproof of this conjecture. As Garey and Johnson conclude [Ref. 9: p.14], even without such a proof, the knowledge that a problem is **NP**-complete suggests, at the very least, that a major breakthrough will be needed to solve it with a polynomial time algorithm.

5. **#P-Complete Problems**

So far the discussion of **NP**-completeness has centered around *decision* problems with yes-no answers. In many cases, however, the real question to be answered goes beyond simply whether a solution exists or not (yes or no). It may be important to find out how *many* solutions there are. Then the problem becomes an *enumeration problem*. For example, associated with the **NP**-complete decision problem 'Does this graph have a Hamiltonian path?' is the enumeration problem 'How many distinct Hamiltonian paths are there in this graph?'

According to Garey and Johnson [Ref. 9: p.167], "Enumeration problems provide natural candidates for the type of problem that might be intractable even if **P** = **NP**." Even if the basic decision problem could be solved in polynomial time, it is not at all clear that the *number* of distinct solutions could be determined in polynomial time. Note that enumeration problems do not require all the solutions to be *displayed*, only counted. Thus the number of Hamiltonian paths in a graph may be exponentially large and an exponential amount of time would be required to list them all, but the answer to the enumeration problem is just a single number. Some enumeration problems can be solved in polynomial time. For example, the question 'Given a graph G, how many Eulerian paths are there for G?' can be solved with a polynomial time algorithm, like the basic decision problem. However, some enumeration problems do not appear to be solvable in polynomial time even though the associated decision problem *can* be solved in polynomial time.

To encompass these considerations, the ideas behind **NP**-complete problems can be extended. A new class, designated ***#P-complete*** can be used to categorize enumeration problems. This is intended to capture the additional difficulty of

enumerating solutions, not just their existence. This class is defined in a way analogous to the class of **NP**-complete problems. Many of the enumeration problems associated with basic **NP**-complete problems are #**P**-complete. What is interesting is that some enumeration problems are now known to be #**P**-complete even though their associated decision problems are *not* known to be members of **NP** [Ref. 9: pp.168-170].

The significance of all this is that if a practical problem can be shown to be #**P**-complete, the search for an efficient, exact algorithm that solves the problem might never be productive. This is not to say that one never will be found, but rather that a major breakthrough will be required. And if an algorithm is ever discovered that solves the problem in *polynomial time*, the implications will be very far reaching. Even though the members of the class of #**P**-complete problems are not yet *provably* intractable, it seems reasonable to operate on the assumption that they are. With that in mind, the investigation of approximation algorithms is certainly of practical significance.

C. EFFICIENCY OF ALGORITHMS FOR PERFORMING THE RANDOMIZATION TEST

1. Randomization Test is an Enumeration problem

Performing the randomization test for two independent samples to accomplish a significance test is an enumeration problem. The remainder of this chapter is devoted to showing that it is #**P**-complete. The fact that it is an enumeration problem is seen by considering the structure of the test. The significance level α is obtained by counting subsets of size n out of $n+m$ elements such that the sum of the elements in each subset is equal to or more extreme than the fixed value $\mathbf{T}_0$. Analogous to the problems discussed in the last section, an associated decision problem could be stated "For some fixed number K, are there K or more subsets of size n for which the sum of the elements is equal to or more extreme than $\mathbf{T}_0$?" This could be answered *yes* or *no* if a value of K were specified beforehand. This would effectively correspond to using the randomization procedure to perform a *hypothesis test*, because the value of K could be determined from the desired value of α_0 using the relation $\alpha_0 = K \binom{n+m}{n}$. In the case of *significance testing*, there is no pre-specified value α_0. To calculate the significance level, we need to know how *many* subsets have sums equal to or more extreme than $\mathbf{T}_0$. In this case, the randomization procedure becomes an enumeration problem rather than a decision problem. The implications of this are discussed in the last chapter.

2. Significance Testing is #P-Complete

To show that using the randomization procedure to perform a significance test is **#P**-complete, two steps are required. First, an enumeration problem is introduced which is known to be **#P**-complete. Second, performing the randomization procedure is shown to be of *equivalent difficulty* to this problem. The **#P**-complete problem is termed the Kth *LARGEST SUBSET* problem and is described next.

The Kth LARGEST SUBSET problem can be stated succinctly using the terminology and format of Garey and Johnson [Ref. 9: p.114]:

Problem Instance: Given a finite set A, a size $s(a) \in \mathbf{Z}^+$ for each $a \in A$, and two nonnegative integers $B \leq \sum s(a)$ and $K \leq 2^{|A|}$.

Question: Are there K or more distinct subsets $A' \subseteq A$ for which the sum of the sizes of the elements in A' does not exceed B?

The notation $|A|$ is defined as the number of elements in the set A. It is not yet known if this decision problem is in the class **NP**, but it is known that the corresponding enumeration problem is **#P**-complete.

Performing the randomization test for two independent samples (assuming significance testing) can be described using the same kinds of set theoretic objects as are used in the Kth LARGEST SUBSET problem above. This can be done in the following way. Let A be the set of $n+m$ elements consisting of the **X** and **Y** observations taken together; that is, $A = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, \mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m\}$. Let the size $s(a)$ be the *positive integer representation* of each element of A. This does not restrict applicability of this result to positive integer observations, however. To show why, consider the following. Note that the test statistic being used $\mathbf{T} = \sum \mathbf{X}_i$ is just the sum of n elements selected from the set A. Suppose some of the elements of A are negative. Then choose a positive constant q such that when q is added to every member of the set A, all the elements will become positive numbers. Every value of the test statistic will also be increased by a constant value, namely nq. This has the effect of shifting the randomization distribution by a fixed amount, and it is obvious that the counting process used to determine the significance level of the test is unaffected.

The next question that might be asked is, what if the elements of A (the **X** and **Y** observations) are real numbers? If the elements in A are the observations from some actual experiment, then any measuring device used can only produce results accurate to within some fixed number of decimal places. Therefore, even though the set of possible measurements is *theoretically* a subset of the real numbers, it in actuality

can only be a collection of integer values over some range; the decimal point is immaterial. Even if real numbers could actually be obtained from some experiment, they would still have to be represented internally in any physical computing device by a fixed number of bits. Again, the position of the decimal point is immaterial; the set of values that can actually be represented is restricted to some collection of integers.

The implication of the preceding paragraphs is that any real experimental data can be thought of as positive integer valued if computing devices are used to perform statistical tests. Thus, any results about algorithmic efficiency stated in terms of positive integers apply whether the true observations are real numbers, integers, or negative values.

Next, let the number B equal the value of the originally observed test statistic T_0. Let K be the number of test statistic values equal to or more extreme than T_0. Then the *enumeration problem* associated with the question "Are there K or more distinct subsets $A' \subseteq A$ for which the sum of the sizes of the elements does not exceed B" is almost equivalent to performing the two sample randomization test. Note that the above question specifies *K or more distinct subsets $A' \subseteq A$*. This includes *all* subsets of A, regardless of how many elements they contain. The number of such subsets is 2^{n+m} since the number of elements in A is $n+m$. In the randomization test, though, we are interested in counting only those subsets with a *fixed* number of elements, namely n. This is equivalent to enumerating all instances where the test statistic value is equal to or more extreme than T_0, since the test statistic is formed by subsets of size n.

Restricting the enumeration problem to subsets of size n is also #P-complete. This can be shown as follows. Suppose we had available an algorithm which could enumerate the number of subsets of size i for which the sum of the sizes of the elements does not exceed B for any fixed value of i such that $0 \le i \le n+m$. Note that by selecting $i=n$, this algorithm would perform the enumeration required for the randomization test. Suppose further that this algorithm operated in polynomial time, that is, in time bounded by a polynomial in $N = n+m$. Then, by simply incrementing i sequentially from 0 to $n+m$ and using this algorithm repeatedly, we could count *all* the distinct subsets $A' \subseteq A$ for which the sum of the sizes of the elements does not exceed B. This is true because of the relationship

$$\sum_{i=0}^{N} \binom{N}{i} = 2^N \qquad \text{(eqn 3.2)}$$

where $N = n+m$. In other words, we could solve the Kth LARGEST SUBSET problem by using our algorithm $N+1$ times. This would mean the time required to enumerate all the subsets $A' \subseteq A$ for which the sum of the sizes of the elements does not exceed B would be bounded by a function of the form $(N+1)p(N)$ - but this is easily seen to be another polynomial. This is a contradiction, because the Kth LARGEST SUBSET problem is **#P**-complete, and its solutions cannot be enumerated in polynomial time. Therefore, any enumeration algorithm that only counts subsets of fixed size n is also **#P**-complete.

IV. ANALYSIS OF AN APPROXIMATE RANDOMIZATION TEST PROCEDURE

A. INTRODUCTION

1. Reasons For Using An Approximate Method

In the previous chapter, it was shown that performing the randomization test for two independent samples is computationally a **#P**-complete enumeration problem when the method is used for significance testing. This means that a fast and efficient way to perform the test is not likely to exist. Certainly one is not known at the present time. In practical terms, the amount of computer time required to complete the necessary calculations becomes totally unreasonable for large data sets. Therefore, if the randomization test is to be used regularly, some way must be found to obtain *approximate* results that are almost as good as the exact results but don't require anywhere near as much computation time.

2. Considerations When Using Approximations

There are many approaches one could take in devising an approximate randomization test. The idea is to come up with a method that yields significance levels very close to those that would result if the exact test were used on the same data. The method should give good results over a wide range of conditions and it should require only a modest amount of computer time. Ideally, it should be possible to establish bounds on the errors involved with using the approximation. These bounds should result from an analytic investigation of both the approximation and the exact test.

Unfortunately, in the case of the randomization test, analytic results are hard to come by. When a randomization test is used, the test statistic can take on as many as $\binom{n+m}{n}$ values. The distribution of these values is called the *randomization distribution*. It is important to note that this is a *conditional* distribution. That is, it is formed by using the *given observations*. Therefore, this distribution changes every time a set of observations is taken. It can easily be shown that the randomization distribution *asymptotically* approaches one of the standard distributions, such as normal or chi-square, but the use of the asymptotic distribution as an approximation may not be accurate in some cases. As Conover [Ref. 4: p.327] indicates, when the observations change from one sample to the next, it is impossible to measure the accuracy of any asymptotic distribution.

Another problem with developing analytic results is that the underlying distributions of the **X** and **Y** populations are not required to be of some specific form. It might be possible to derive error bounds on a conditional basis. That is, by stating something like *'If the underlying distributions are of the forms $F(x)$ and $G(y)$, then the maximum error incurred by using this approximation is $H(x, y)$.'* Of course, the number of possible distributions is infinite, and the 'true' underlying distributions can never be known with certainty, so this approach may have limited value.

3. Method Studied

There are several ways that have been used to perform approximate randomization tests. One way is to simply use the standard t-test, even though the data may not be normally distributed, and then hope that the results are not too far off. Other methods have involved using only portions of the data, sampling from the total number of combinations, and fitting various distributions. Some of these methods are briefly described in the next section. The method studied here with the aid of simulation is the *2-moment fit* method. Significance levels obtained with this method are compared to those obtained from the exact randomization test and the t-test. Power curves for each test are also generated as a separate indicator of performance.

B. PERFORMING APPROXIMATE RANDOMIZATION TESTS

1. Subsampling

One way to perform an approximate randomization test is to determine the significance level from a subset of the test statistic values making up the randomization distribution. The subset consists of combinations chosen at random from the $\binom{n+m}{n}$ combinations possible. The test statistic values are computed for these combinations only and an approximate significance level is obtained. This is called *subsampling* and the combinations can be selected through random sampling with replacement or without replacement. For example, if an experiment yielded 30 **X** observations and 30 **Y** observations ($n = m = 30$), the total number of test statistic values making up the randomization distribution would be $\binom{60}{30}$, which is about 1.18×10^{17}. Instead of comparing all those combinations to the original test statistic value $\mathbf{T}_0$, a much smaller set of test statistic values, say a few thousand, could be formed from combinations selected at random out of the $\binom{60}{30}$ available. This smaller number of test statistic values could then be compared to $\mathbf{T}_0$ and an approximate significance level could be found.

Subsampling is a very attractive approximation method, since even a few thousand test statistic values can be generated at random and compared rather quickly. The method has intuitive appeal also, because every combination out of the $\binom{n+m}{n}$ possible is considered equally likely if the null hypothesis is true. Sampling from a set of equally likely objects should yield representative subsets. The only questions to be answered are how large a sample is required and whether sampling should be done with or without replacement. Studies done on this subject [Ref. 11: pp.43-45] make it appear that sampling with replacement is acceptable and the use of sample sizes as small as 1000 can provide good results.

2. **Blocks**

Another approximation method, which is a variation on the subsampling scheme, is the use of *blocks*. This method can be applied to the randomization test for two independent samples in the following way, which is described by Boyett and Shuster [Ref. 12: p.666]. Within the **X** and the **Y** samples, an appropriate number of blocks is formed by random allocation, each block having the same number of observations. Then an exact randomization test is used on the *block sums*. For example, if the data consisted of 30 **X** observations and 30 **Y** observations, six blocks of five observations each could be randomly formed within the **X**'s and the **Y**'s. The sum of the observations in each block would be found. Then the exact randomization procedure could be used on the block sums. The number of all such sums would only be $\binom{12}{6} = 924$. Again, significant savings could be achieved over performing the exact test on all 1.18×10^{17} combinations of the observations without blocking.

How many blocks should be chosen depends on how accurate the results need to be and on how much computation time is considered acceptable. Using a small number of blocks may be less accurate, but the computer time required will certainly be less than if many blocks are used. It should also be noted that it may not be possible to form a convenient number of blocks (all containing the same number of observations) without discarding some of the data. For example, if we had 23 **X** observations and 26 **Y** observations, we might form 7 blocks of 3 observations each within the **X**'s and 8 blocks of 3 observations each within the **Y**'s. In this case, two **X** and two **Y** observations would have to be discarded.

3. **T-test as an Approximation**

If *random* sampling from *normal* distributions can be assumed, the standard two sample t-test is the appropriate parametric procedure that can be used to perform

a comparison of means. However, even if the underlying distributions are not normal, a histogram of the test statistic values from the randomization procedure often resembles a bell-shaped normal density. This is true for the test statistic being used here, namely $T = \sum X_i$. An equivalent test statistic that yields the same results is $T = (\sum X_i)/n - (\sum Y_j)/m$. If this test statistic is used, a histogram of the test statistic values is centered at the origin and takes on the appearance of a central t density. In fact, the randomization distribution arising from the use of this test statistic is usually approximated reasonably well by an appropriately scaled t distribution. Hence, as Box, Hunter and Hunter [Ref. 2: pp.95-97] observe, provided that a *randomized experiment* is performed, t-tests can be used as approximations to exact randomization tests even if the underlying distributions are not normal.

4. 2-Moment Fit Method

The next approximation method to be discussed will be called the *2-moment fit* method. The basic principle involved is simply that of using a continuous distribution to approximate a discrete distribution. As mentioned in the last section, if histograms of the true randomization distribution are examined, it becomes apparent that in many

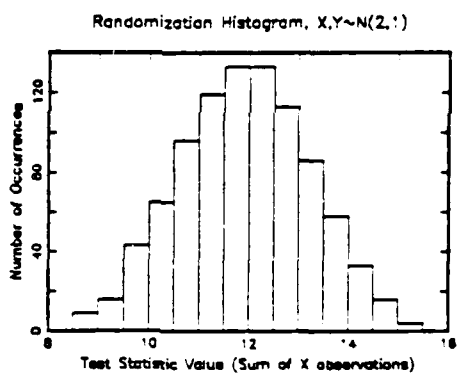


Figure 4.1 Typical Randomization Histogram.

cases the histograms seem to have a characteristic bell shape as in Figure 4.1. In fact, if the null hypothesis is true, the distribution of the randomization test statistic should asymptotically approach a normal distribution under easily met conditions [Ref. 4: p.327]. With this in mind, it seems reasonable to assume that a normal distribution with some mean μ and standard deviation σ might be *fitted* to the randomization distribution, as shown in Figure 4.2.

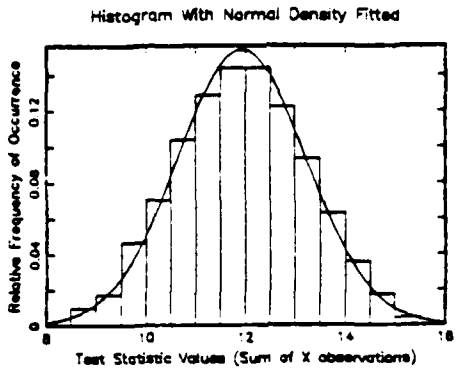


Figure 4.2 Normal Density Fitted to Randomization Histogram.

If the normal density 'fits' reasonably well, the area under a given portion of the curve should approximate the corresponding area under the randomization histogram bars. The area represented by the histogram bars equal to or more extreme than the originally observed test statistic value T_0 corresponds to the significance level

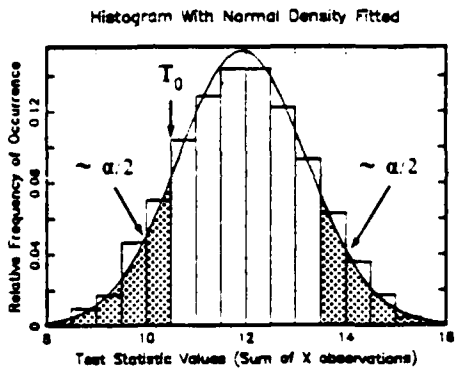


Figure 4.3 Tail Areas Correspond to α.

α of the test. This is shown in Figure 4.3. Therefore, the laborious exact calculation of α by enumeration can be replaced by fitting an appropriate normal curve and using tables to find the required areas. The approximate α obtained in this manner could be very quickly calculated, no matter how large the number of test statistic combinations.

To 'fit' a normal distribution to the distribution of test statistic values, the first two moments (functions of μ and σ) of the normal distribution must be related to two values in the test statistic distribution, hence the name '2-moment fit'. Two values that could be chosen are simply the end points - that is, the smallest and the largest

values that the test statistic takes on when the randomization test is actually performed. It is easy to find these two values without enumerating all possible combinations; just sum the n smallest and then the n largest observations from the combined set $\{\mathbf{X},\mathbf{Y}\}$. Once the smallest and largest test statistic values are found, the probabilities associated with their occurrence are easily determined from knowledge of the total number of test statistics possible, which is $\binom{n+m}{n}$. These probabilities are used to find the $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ that completely describe the fitted normal density. For a full derivation of the equations involved with this method, see Appendix A.

The 2-moment fit method seems intuitively appealing. As the number of test statistic values that make up the randomization distribution gets large, it seems reasonable to expect that a continuous function (the normal distribution) should more closely approximate the true discrete distribution. The degree to which the approximation yields values 'close' to the true $\boldsymbol{\alpha}$ also depends on the degree to which the normal curve follows the *shape* of the true discrete distribution.

C. A COMPARISON OF EXACT AND APPROXIMATE METHODS USING SIMULATION

1. Purpose of Simulation

The inherent difficulties associated with deriving *analytic results* describing error bounds have been discussed previously. Because of these difficulties, the errors that result from the use of approximate methods can be studied conveniently using simulation techniques. After a simulation has been run several times, approximate error bounds can be established and confidence limits on those bounds can be applied. A variety of input conditions and underlying distributions can be entered, and the effect of each can be analyzed.

A simulation was written for the sole purpose of comparing the *significance level* and *power* of the exact randomization test under varying conditions to two alternative methods:

(1) The 2-moment fit approximation

(2) The two-sample t-test.

A complete description of the simulation and an interpretation of the major results obtained from it are the subjects of the remainder of this chapter.

2. Description of Simulation

a. Overall Structure

The overall structure of the simulation can be outlined as follows. The purpose is to compare the power and significance level of the exact randomization test to the 2-moment fit approximate method and the standard t-test. This is accomplished by repeatedly generating sets of **X** and **Y** observations from preselected distributions. The parameters of the **X** and **Y** distributions can be independently varied. At each repetition, a *significance test* is performed on the hypothesis

$$H_0: \boldsymbol{\mu}_x = \boldsymbol{\mu}_y$$
$$\text{vs. } H_1: \boldsymbol{\mu}_x \neq \boldsymbol{\mu}_y$$

using each of the three methods. The results are recorded in a file for analysis by separate means. To generate power curves for each test method, the parameters of the **X** distribution are held fixed while the mean of the **Y** distribution is varied over a specified range. Using a pre-selected value denoted $\boldsymbol{\alpha}_0$, the probability of rejecting the null hypothesis when it is false (the definition of power) is empirically determined for each difference $\boldsymbol{\mu}_x - \boldsymbol{\mu}_y$ in the specified range.

The following basic distributions can be selected for the **X** and **Y** samples:

(1) Normal

(2) Exponential

(3) Uniform

All input parameters can be varied. These include:

(a) Mean of **X** and **Y** distributions (all types)

(b) Standard deviation of **X** and **Y** distributions (normal)

(c) Sample sizes n and m

(d) Number of repetitions

(e) Range over which power curves are to be generated

(f) Value of $\boldsymbol{\alpha}_0$ to use in obtaining power values.

b. Programming Details

The simulation was programmed in VS FORTRAN. Routines in both the IMSL and NON-IMSL libraries were utilized for random number generation and calculation of values associated with the normal and t distributions. A complete program listing is provided in Appendix B.

3. **Results and Interpretation**

a. Significance Levels

The first studies conducted with the simulation were those in which the significance levels yielded by each of the three methods were compared. These comparisons were made by generating n **X** observations and m **Y** observations from the same type of distribution, except that μ_x and μ_y could each be varied. Once a set of **X** and **Y** observations was generated, all three tests (exact randomization, 2-moment fit approximation, and the t-test) were performed on that set and the three resulting significance levels were recorded. This process was repeated a selectable number of times. The following input conditions were varied:

(1) Distribution type (Normal, Exponential, and Uniform)

(2) H_0 true ($\mu_x = \mu_y$) and H_0 false ($\mu_x \neq \mu_y$)

(3) Sample sizes n and m for **X** and **Y** sets, respectively

(4) Number of repetitions

Sample sizes up to $n = 11$ and $m = 11$ were examined, and up to 200 repetitions were used. For each input condition, the significance levels from the 2-moment fit method and the t-test were plotted against the corresponding values obtained from the exact randomization test.

The plotted data from these simulation runs indicated that for all sample sizes larger than $n = 4$ and $m = 4$, the 2-moment fit method generally produced **smaller** significance values than either the exact randomization test or the t-test, with a maximum average error of about 0.2 units of probability. The significance values obtained from the t-test were much closer to those from the exact randomization test. This behavior was observed for all three distributions and for every combination of input parameters. Example plots appear in Figures 4.4 and 4.5.

For simulated sample sizes less than $n = 4$ and $m = 4$, the fact that the randomization test can only produce a discrete set of significance values tended to introduce more variability in the results. It was also noticed that the 2-moment fit method yielded essentially the same values as the other two methods when the significance levels were close to either 0 or 1.

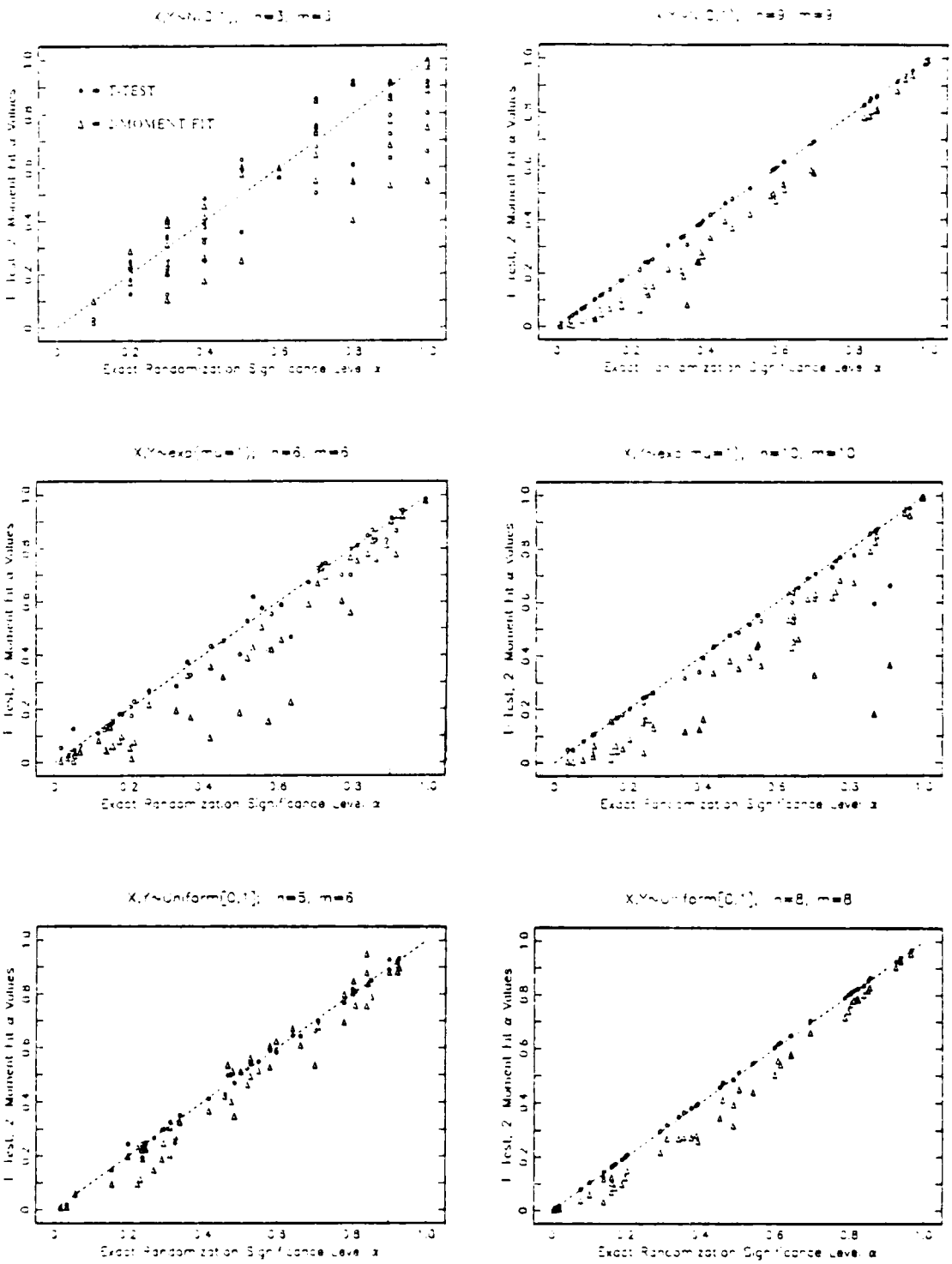


Figure 4.4 Significance Level Comparisons: H_0 True.

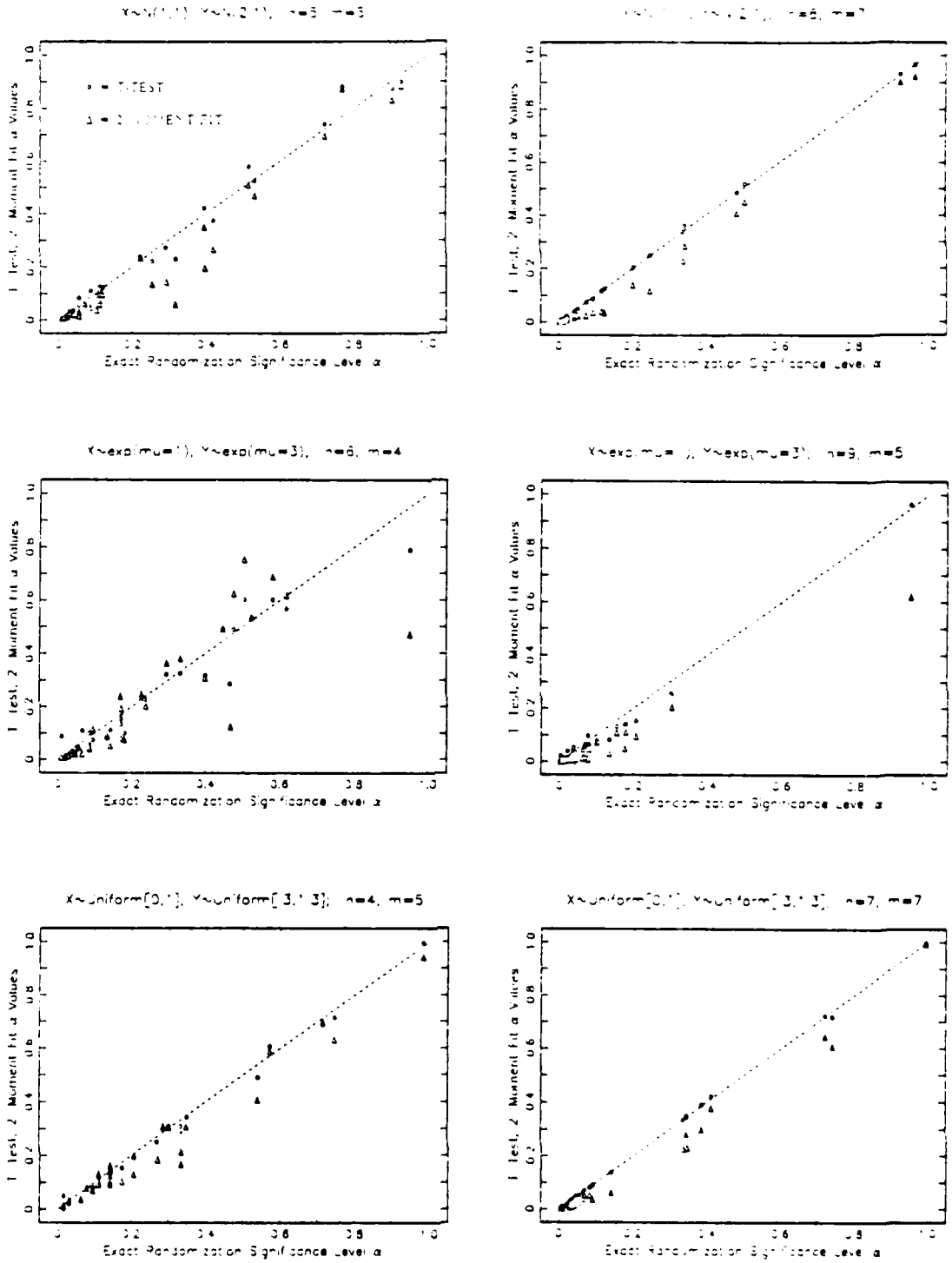


Figure 4.5 Significance Level Comparisons: H_0 False.

b. *Power Curves*

To develop power curves, the simulation was run in a manner similar to the significance testing situation. The same input conditions were varied, except that a value of α_0 was specified beforehand. Each time a simulated significance level occurred that was less than α_0, the test that produced that level [illegible] the null hypothesis H_0. The number of times each test rejected H_0 was then divided by the number of repetitions to yield average power values. [illegible] a range of μ_y values around a fixed value of μ_x.

Examples of power curves generated from the simulation appear in Figure 4.6. It appears from the power curves that the 2-moment fit method rejected the null hypothesis *too* often. That is, the power curves for this method were artificially high. This follows from the fact that the 2-moment fit method generally yielded significance values that were too low. When the null hypothesis is true ($\mu_x = \mu_y$), the power curve should pass through the selected value of α_0. This was not the case for the 2-moment fit curves; they were consistently too high.

The power of the t-test was close to the power of the exact randomization test for runs involving the normal and the uniform distributions. However, for the exponential distribution, the exact randomization test power curve was always above the power curve for the t-test. This is consistent with the theoretical results discussed in Chapter Two - namely that the randomization test is the *uniformly most powerful test* against the subclass of alternatives that includes the exponential densities.

c. *Randomization Histograms*

Randomization distribution histograms were plotted for many of the input conditions used in the simulation. Most of these were unimodal in appearance, as expected. However, some of the histograms resulting from runs involving the exponential distribution were *multimodal* when the null hypothesis was false. For two examples, see Figure 4.7. This behavior could help explain why the 2-moment fit method does not approximate the true significance level very well in these cases. The 2-moment fit method tries to fit a (unimodal) normal density to the test statistic values, and if those values exhibit multimodal tendencies, large errors are likely.

4. Summary of Results

The most significant results obtained from the simulation are (1) the t-test is a good approximation to the exact randomization test in most cases, and (2) the 2-moment fit method usually yields smaller significance values than either the

$(n=4,\ m=4)$ $(n=6,\ m=6)$

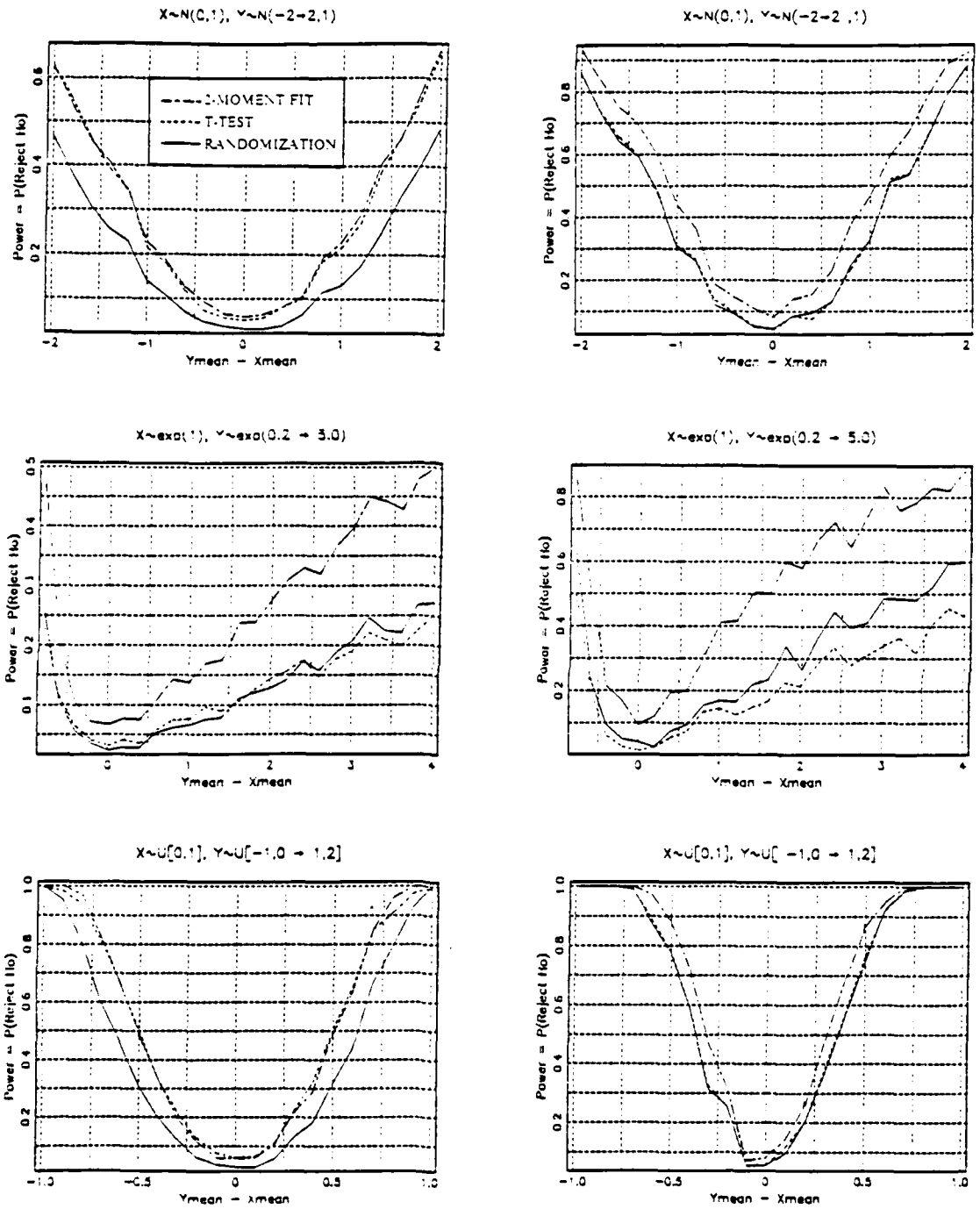


Figure 4.6 Example Power Curves For α_0 = .05.

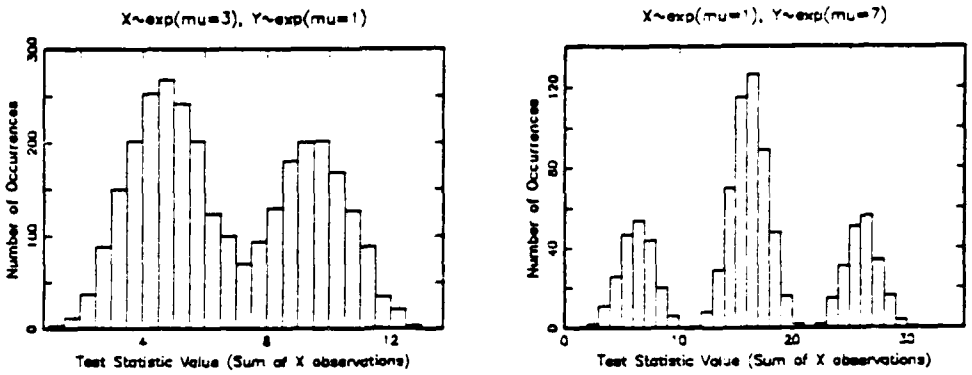


Figure 4.7 Multimodal Histograms.

randomization test or the t-test. The maximum average error incurred is about 0.2 units of probability. The reason the 2-moment fit method does not work very well is probably related to its use of the two most extreme values of the test statistic. Finally, the power curves that were generated showed that the randomization test can be more powerful than the t-test when samples are exponentially distributed.

V. SUMMARY

A. MAJOR RESULTS AND CONCLUSIONS

This thesis has addressed issues related to the practical implementation of the randomization test for two independent samples. The test was described as a method for comparing the means of two populations **X** and **Y** from which independent samples have been drawn. The method can be categorized as a *nonparametric* statistical procedure because assumptions about the specific form of the **X** and **Y** distributions and associated parameters are not necessary.

Although many nonparametric procedures are 'weaker' than corresponding parametric techniques, the randomization test is at least as good from a theoretical standpoint as its parametric counterpart, the t-test. In some cases, its performance can be better. Some of the indicators of a good statistical test are efficiency, unbiasedness and power. The randomization test has been shown to have an asymptotic relative efficiency of 1.0, it is an unbiased test, and it is the uniformly most powerful test in certain situations. Each of these results obtained from the literature was discussed. The implication is that the randomization test should be the preferred method of testing equality of means unless reasonable justification exists for the use of the t-test (normality assumptions can be supported, for example).

Even though the randomization test may be the best way to compare population means in theory, it can be so time-consuming to actually perform the test on a computer that it is not often used unless sample sizes are relatively small. The structure of the test is basically a counting procedure involving combinations of the **X** and **Y** observations. As the number of observations increases, the number of possible combinations becomes so huge that even the fastest computing machinery cannot perform the test in a realistic amount of time. There is no known way to perform the test efficiently for large sample sizes in the general case.

To be more specific about what is computationally efficient and what is not, topics from the theories of **NP**-complete and **#P**-complete problems were introduced in this thesis. Algorithms for performing tasks or solving problems on a computer can be broadly classed as efficient if they can be executed in *polynomial time*. If a problem can be classified as **NP**-complete or **#P**-complete, it is extremely unlikely that a polynomial

time algorithm exists which can solve it. The randomization test for two independent samples was shown to be a #P-complete enumeration problem when significance testing is being performed. Therefore, an efficient algorithm for implementing the test on a computer is not likely to exist.

Because of the problem of excessive computation time, ways have been sought to perform the randomization test *approximately*; that is, to obtain significance values close to those that would result if the exact test were used on the same data. Some of the ways that have been suggested to perform an approximate test include subsampling, the use of blocks, asymptotic distributions or simply using the standard t-test. There are advantages and disadvantages involved with the use of each of these methods.

Another way to perform an approximate test is to fit a normal distribution to the distribution of test statistics that would result if the exact procedure were used. This method extracts the largest and the smallest test statistic values and uses them to find the first two moments of the fitted normal distribution, hence the name *2-moment fit method*. This method was studied with the aid of a simulation. The simulation compared the performance of the 2-moment fit approximation to the exact randomization test and the standard t-test. Significance levels were found and power curves were developed for each test under varying conditions.

Several conclusions could be drawn from analyzing the simulation data. The first conclusion is that the 2-moment fit method will, in general, *underestimate* the true significance level that would result from using the exact randomization test. This behavior occurred for all conditions studied in the simulation, which included changes in the sample distributions, changes in location parameters, and both true and false null hypotheses. The maximum average error resulting from the use of the 2-moment fit method approximation when the null hypothesis is *true* is about 0.2 units of probability. Error in this context is defined to be the true significance level from the randomization test minus the approximate significance level.

Another important conclusion is that the t-test is quite adequate as an approximation to the exact randomization test in most cases. Statements to this effect are in the literature, and the simulation results proved to be consistent. The significance values produced by the t-test were generally very close to those obtained from the exact randomization test. When power curves were developed, though, it was demonstrated that the exact randomization test can be more powerful than the t-test

when the underlying distributions are exponential. This is also consistent with theoretical results identifying the randomization test as the uniformly most powerful test in that particular situation.

The overall conclusion of this thesis is that the randomization test for two independent samples should be used in its exact form for testing equality of means if sample sizes are small and there is concern over whether or not assumptions of normality can be justified. When sample sizes become large enough that performing an exact randomization test requires more than a reasonable amount of time, the t-test provides good approximate results. Of course, the t-test is always the most appropriate test to use in the first place if one is willing to assume normality actually exists.

B. AREAS FOR FURTHER RESEARCH

1. Approximations

Approximate methods appear to be the most practical ways to implement randomization tests if they are desired for large sample sizes. More research in this area could be of value. It might even be possible to obtain more accuracy from the 2-moment fit method in some way. But this research indicates that a significant improvement would be required before the method could be considered better than the t-test as an approximation.

2. Pseudo-Polynomial Time Algorithms

One area of research that could prove to be very significant would be the development of a *pseudo-polynomial time* algorithm to perform the randomization test when hypothesis testing is being done. A pseudo-polynomial time algorithm is one that can be executed in polynomial time if a bound on the allowable input lengths is established ahead of time. For a more detailed explanation, see Garey and Johnson [Ref. 9: p.91]. This would correspond to selecting upper limits for the sample sizes n and m and designing an algorithm based on the knowledge that larger sample sizes will not be input to the routine. An example of this kind of approach is the use of *dynamic programming* to solve the classic knapsack problem.

It was shown in Chapter Three that using the randomization test to perform *significance testing* is a **#P**-complete enumeration problem. However, if *hypothesis testing* is being performed, the test is really a decision problem with a *yes* or *no* answer. If maximum allowable sample sizes were to be established in advance, an approach similar to dynamic programming might be used to solve the problem much more efficiently than using total enumeration. An indication of how this could be applied

appears in Garey and Johnson [Ref. 9: pp.90-92]. If a suitable algorithm could be designed that runs quickly in practice (even if it is theoretically a pseudo-polynomial time algorithm), an important step would be made toward more widespread use of randomization tests for statistical hypotheses.

APPENDIX A
DERIVATION OF THE 2-MOMENT FIT METHOD

Purpose: To obtain approximate significance values by fitting a normal distribution to the distribution of exact randomization test statistics. Areas under the resulting normal curve correspond to the proportion of test statistics equal to or more extreme than the originally observed value T_0.

Step 1: Find μ and σ that define a fitted normal density function.

Recall that the test statistic is just the sum of the **X** observations:

$$T = \sum X_i \quad , i = 1, \ldots ,n.$$

Let n be the number of **X** observations and m be the number of **Y** observations. Let T_s be the smallest test statistic value. This value can easily be found by summing the n smallest observations from the combined set {**X**,**Y**}. The *combined set* {**X**,**Y**} is the set of all the **X** and **Y** observations taken together. Similarly, let T_b be the largest test statistic value, which can be found by summing the n largest observations from the set {**X**,**Y**}.

It is possible that either T_s or T_b (or both) are not unique. More than one test statistic value might be the smallest, for example. This could happen if the number of observations is small or there are many ties. To account for this possibility, define the numbers j_s and j_b in the following way:

$$j_s = \text{number of smallest test statistic values}$$
$$j_b = \text{number of largest test statistic values.}$$

One way to determine the numbers j_s and j_b is as follows. Order the set {**X**,**Y**} of $n+m$ observations from smallest to largest. Look at the observation in position n. If it is unique, then T_s is unique and $j_s = 1$. If the observation in position n is not unique, then T_s is not unique. Assume there are k observations that are equal to the observation in position n. Also assume that the k equal observations begin in position $n-r+1$. Then $j_s = \binom{k}{r}$. The number j_b is determined similarly, except the set {**X**,**Y**} must be looked at in the opposite direction, from largest to smallest.

Once $\mathbf{T}_s$, $\mathbf{T}_b$, j_s and j_b have been found, the two extreme points of the randomization distribution are defined. A normal density function is fitted by matching its *tail areas* to the probabilities of randomly selecting the values $\mathbf{T}_s$ and $\mathbf{T}_b$ out of all the $\binom{n+m}{n}$ test statistics available. Let p_s be the probability of selecting $\mathbf{T}_s$ and let p_b be the probability of selecting $\mathbf{T}_b$. Then $p_s = j_s \binom{n+m}{n}$ and $p_b = j_b \binom{n+m}{n}$.

Next, let $\boldsymbol{\Psi}$ represent an arbitrary random variable that is normally distributed with mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$, and let ψ be its density function. To match the tail areas of the function ψ to the probabilities p_s and p_b, set

$$P(\boldsymbol{\Psi} \leq \mathbf{T}_s) = p_s$$

for the lower tail area, and

$$P(\boldsymbol{\Psi} \geq \mathbf{T}_b) = p_b ,$$

which is equivalent to

$$P(\boldsymbol{\Psi} \leq \mathbf{T}_b) = 1 - p_b$$

for the upper tail area. Letting $\mathbf{Z}$ represent a standard normal random variable, the above probabilities can be rewritten in terms of the standard normal distribution function by subtracting $\boldsymbol{\mu}$ and dividing by $\boldsymbol{\sigma}$:

$$P(\boldsymbol{\Psi} \leq \mathbf{T}_s) = P(\mathbf{Z} \leq \frac{\mathbf{T}_s - \boldsymbol{\mu}}{\boldsymbol{\sigma}}) = p_s$$

$$\text{and } P(\boldsymbol{\Psi} \leq \mathbf{T}_b) = P(\mathbf{Z} \leq \frac{\mathbf{T}_b - \boldsymbol{\mu}}{\boldsymbol{\sigma}}) = 1 - p_b .$$

Let z_s be the percentile of the standard normal distribution associated with the probability p_s. That is, $P(\mathbf{Z} \leq z_s) = p_s$. Similarly, let z_b be the percentile associated

with the probability $1 - p_b$. Then

$$z_s = \frac{T_s - \mu}{\sigma}$$

$$\text{and } z_b = \frac{T_b - \mu}{\sigma}.$$

Multiplying through by σ and rearranging yields the two equations

$$\mu + z_s\sigma = T_s$$

$$\mu + z_b\sigma = T_b$$

The above system of linear equations can be easily solved for the quantities μ and σ by standard methods to yield

$$\mu = \frac{z_b T_s - z_s T_b}{z_b - z_s}$$

$$\text{and } \sigma = \frac{T_b - T_s}{z_b - z_s}.$$

These are the values of μ and σ that define the fitted normal density function ψ.

Step 2: Relate the area under the fitted normal density function to the proportion of test statistics equal to or more extreme than T_0.

Two cases must be considered, depending on whether T_0 is in the upper or lower tail of the distribution of all $\binom{n+m}{n}$ test statistics.

Case 1: T_0 is in the lower tail.

Let α be the significance level that would result if an exact randomization test were to be performed on the same data. Recall that α is found from the proportion of test statistics whose values are equal to or more extreme than the originally observed value T_0. This proportion is doubled to yield the value of α because the test is two

tailed. The proportion of test statistics whose values are equal to or more extreme than T_0 is approximately the same as the area under the fitted normal density function ψ to the *left* of T_0, since T_0 is in the lower tail. Since areas under a normal density function correspond to probabilities, the following relation holds:

$$\alpha \sim 2P(\Psi \leq T_0) = 2P(Z \leq \frac{T_0 - \mu}{\sigma})$$

where Z is the standard normal random variable. Substituting the values of μ and σ for the fitted density ψ in the equation yields

$$\alpha \sim 2P(Z \leq \frac{T_0 - \frac{z_b T_s - z_s T_b}{z_b - z_s}}{\frac{T_b - T_s}{z_b - z_s}})$$

Which simplifies to

$$\alpha \sim 2P(Z \leq \frac{z_b(T_0 - T_s) + z_s(T_b - T_0)}{T_b - T_s}) .$$

The probability on the right can be found by consulting a table of the standard normal probability function.

Case 2: T_0 is in the upper tail.

The same reasoning used in Case 1 is applicable. Due to the symmetry of the normal distribution, $P(Z \geq \zeta) = P(Z \leq -\zeta)$ for all ζ. Therefore, the resulting approximation formula for α is the same as in Case 1 with the exception of a minus sign:

$$\alpha \sim 2P(Z \leq -\frac{z_b(T_0 - T_s) + z_s(T_b - T_0)}{T_b - T_s}) .$$

APPENDIX B
SIMULATION PROGRAM LISTING

```
C
C     Principal Variable listing:
C        A()........ Output vector used in combinatorial procedure
C        ALPHA...... Significance level for power curves
C        APOWER..... Approximate power
C        APPROX..... Approximate significance level
C        DELTA...... Difference in means, X and Y
C        DTYPE...... Distribution type
C        DXY()...... Data X and Y vector
C        EPOWER..... Exact power from randomization test
C        EXACT...... Exact significance level
C        ISEED...... Random number generation seed
C        KDX........ Unequal sample size variable(X)
C        KDY........ (Same)(Y)
C        L.......... Largest value of the test statistic
C        NCOMB(,)... No. of combinations
C        NX......... No. of X's
C        NY......... No. of Y's
C        S.......... Smallest value of the test statistic
C        TOLER...... Tolerance on equality of sums
C        TPOWER..... T-test power value
C        TVAL....... T-test significance level
C        ZL......... Quantiles of the standard normal distribution
C        ZS.........  associated with the largest and smallest
C                     values of the test statistics.
C
C ********************************************************
C          Program Begins Here.
C
C
C
      INTEGER NCOMB(2:15,2:15),DTYPE,H,A(15),RESP
      REAL*4 DXY(30),SN(30),E(30),U(30),L
      REAL*8 Q,X,ZS,ZL,Y,P,APPROX
      LOGICAL MTC
C
C   Read in no. of combinations from external file:
C
      READ(UNIT=7,FMT='(I10)',ERR=1) ((NCOMB(I,J),I=2,15),J=2,15)
      GO TO 2
    1 PRINT *,'ERROR IN READ.'
      STOP
    2 PRINT *,'Enter the following parameters:'
      PRINT *,'Alpha'
      READ *,ALPHA
C
      PRINT *,'Distribution type:'
      PRINT *,'     1 = Normal'
      PRINT *,'     2 = Exponential'
      PRINT *,'     3 = Uniform'
      READ *, DTYPE
C
      PRINT *,'Xmean'
      READ *, XMEAN
C
      PRINT *,'Xsigma'
      READ *, XSIGMA
C
      PRINT *,'Ysigma'
      READ *, YSIGMA
C
```

```
      PRINT *,'Ymean range in the form YMIN, YMAX'
      READ *, YMIN,YMAX
C
      PRINT *,'No. of divisions to divide mean range (MDIV)'
      READ *, MDIV
C
      PRINT *,'No. of repetitions (NREPS)'
      READ *, NREPS
C
      PRINT *,'Range of sample sizes: KMIN,KMAX'
      READ *, KMIN,KMAX
C
      PRINT *,'For unequal sample sizes, enter KDX, KDY:'
      READ *, KDX,KDY
C
      ISEED=47771
      TOLER=1.0E-5
C
      YSTEP= (YMAX - YMIN)/MDIV
      IF (YMAX.EQ.YMIN) MDIV=0
C
C   *** Begin Main outer loop: vary mean of Y while holding X fixed.
C
      DO 7003 MSTEP=0,MDIV
C
      YMEAN = YMIN + YSTEP*MSTEP
      DELTA = YMEAN - XMEAN
C
C   ** Next loop: vary sample sizes for X and Y.
C
      DO 7002 KSIZE = KMIN,KMAX
C
C    '(Start with equal sample sizes, then vary by KDX,KDY):
      NX = KSIZE + KDX
      NY = KSIZE + KDY
C
      NC = NCOMB(NX,NY)
      N = NX + NY
      K = NX
C
C  'Initialize power curve counters:
      KE = 0
      KA = 0
      KT = 0
C
C   * Start iteration loop:
C
      DO 7001 ITER = 1,NREPS
C
C   'Select appropriate distribution
C
      GO TO(101,102,103) DTYPE
 101  CALL NORMAL( ISEED,NX,NY,XMEAN,XSIGMA,YMEAN,YSIGMA,DXY )
C
C///'Data import facility:
C     READ(UNIT=10,FMT='(F5.1)') (DXY(I),I=1,12)
C///
          GO TO 200
 102  CALL EXPONL( ISEED,NX,NY,XMEAN,YMEAN,DXY )
          GO TO 200
 103  CALL UNIFRM( ISEED,NX,NY,XMEAN,YMEAN,DXY )
C
C   'Find value of observed test statistic T0:
C
 200  T0 = 0.
       DO 210 IX = 1, NX
 210  T0 = T0 + DXY(IX)
C
C   'The following section performs an exact randomization test of
C    significance for the null hypothesis Ho: Xmean = Ymean against
```

```
C     a two sided alternative.  The sum of the X observations is used
C     as the test statistic.  In addition, the largest and smallest
C     test statistic values are found for later use in the approximate
C     method.
C
C    'Initialize parameters/ counters:
C
        S = T0
        L = T0
        JS = 0
        JL = 0
        NLE = 0
        NGE = 0
C
C    'Generate all possible combinations of the elements in DXY()
C     taken NX at a time:  This algorithm is given in Nijenhuis, A. &
C     Wilf, H.S.,  Combinatorial Algorithms for Computers and Cal-
C     culators, 2nd Ed.   Academic Press, 1978 ,  pp. 32-33.
C
        MTC = .FALSE.
        DO 400 KC = 1, NC
        IF (MTC) GO TO 40
        M2=0
        H=K
        GO TO 50
   40 IF (M2.LT.N-H) H=0
        H=H+1
        M2=A(K+1-H)
   50 DO 51 J=1,H
   51 A(K+J-H)=M2+J
        MTC=A(1).NE.N-K+1
C
C    'Find sum of the X's for this combination:
C
        T = 0.
        DO 300 IL = 1, NX
 300    T = T + DXY ( A(IL) )
C
C///'Test statistic output facility:
C       WRITE(10,69) T,NC
C 69    FORMAT(F10.4,2X,I6)
C///
C    'Find smallest & largest sums and count them:
C
        IF ( ABS(T-S) .LT. TOLER ) THEN
            JS = JS + 1
            GO TO 310
        ELSE IF ( T .LT. S ) THEN
            S = T
            JS = 1
        END IF
C
 310    IF ( ABS(L-T) .LT. TOLER ) THEN
            JL = JL + 1
            GO TO 320
        ELSE IF ( T .GT. L ) THEN
            L = T
            JL = 1
        END IF
 320    CONTINUE
C
C    'Count # of observations <= and >= T0:
        IF ( T.LE.T0 ) NLE = NLE + 1
        IF ( T.GE.T0 ) NGE = NGE + 1
C
 400    CONTINUE
C
C
C    'Compute exact significance level:
        IF ( NLE.LE.NGE ) EXACT = REAL(2*NLE)/REAL(NC)
```

```
      IF ( NGE.LE.NLE ) EXACT = REAL(2*NGE)/REAL(NC)
C
C   'Perform approximate method:
C
      Q = DBLE(JS) / DBLE(NC)
      CALL INORM ( Q,X,IERR )
C
      IF ( IERR.EQ.1 ) THEN
         PRINT *,'Error in subroutine INORM'
         STOP
      END IF
C
      IF ( JS.EQ.JL ) THEN
         ZS = X
         ZL = -X
      ELSE
         ZS = X
         Q = DBLE(JL) / DBLE(NC)
         CALL INORM ( Q,X,IERR )
         IF ( IERR.EQ.1 ) THEN
               PRINT *,'Error in subroutine INORM (2)'
               STOP
         END IF
C
         ZL = -X
      END IF
C
      Y = ( ZL*(T0-S) + ZS*(L-T0) ) / (L-S)
      CALL MDNORD ( Y,P )
      IF ( P.LE. 0.5D0 ) THEN
         APPROX = 2.0D0 * P
      ELSE
         APPROX = 2.0D0 * ( 1.0D0 - P )
      END IF
C
C
C   'Perform standard t-test:
C
      CALL TTEST ( DXY,NX,NY,TVAL )
C
C   'Increment power curve generators:
C
      IF ( EXACT.LE.ALPHA ) KE = KE + 1
      IF ( APPROX.LE.ALPHA ) KA = KA + 1
      IF ( TVAL.LE.ALPHA )  KT = KT + 1
C
      WRITE(8,1000) DTYPE,XMEAN,YMEAN,DELTA,NX,NY,TVAL,EXACT,APPROX
 1000 FORMAT(I1,3(2X,F6.3),2(1X,I3),3(2X,F7.5))
C
 7001 CONTINUE
C
C
C   'Calculate ave. power values for this sample size:
C
      REPS = REAL(NREPS)
      EPOWER = KE / REPS
      APOWER = KA / REPS
      TPOWER = KT / REPS
C
C   'Write power curve values into separate file:
C
      WRITE(9,2000) DTYPE,NX,NY,DELTA,EPOWER,APOWER,TPOWER
 2000 FORMAT(3(I3,2X),F6.3,3(2X,F7.5))
C
 7002 CONTINUE
 7003 CONTINUE
C
C
      PRINT *,' Another run?  0 = no,  1 = yes'
      READ *, RESP
```

```
      IF ( RESP.EQ.1 ) GO TO 2
C
      STOP
      END
C
C
C
C
      SUBROUTINE NORMAL ( ISEED,NX,NY,XMEAN,XSIGMA,YMEAN,YSIGMA,DXY )
C
C   'Generates normal X and Y samples.
C
      DIMENSION DXY( NX + NY ),SN(30)
      NGEN = NX + NY
      CALL SNOR ( ISEED,SN,NGEN,2,0 )
C
C   'Generate X:
      DO 1 IX = 1, NX
  1   DXY(IX) = XMEAN + XSIGMA * SN(IX)
C
C   'Generate Y:
      DO 2 IY = 1, NY
  2   DXY(NX+IY) = YMEAN + YSIGMA * SN(NX+IY)
C
      RETURN
      END
C
C
C
C
      SUBROUTINE EXPONL( ISEED,NX,NY,XMEAN,YMEAN,DXY )
C
C   'Loads DXY with exponentially distributed X,Y.
C
      DIMENSION DXY( NX+NY ),E(30)
      NGEN = NX + NY
      CALL SEXPN ( ISEED,E,NGEN,2,0 )
C
C   'Generate X:
      DO 1 IX = 1, NX
  1   DXY(IX) = XMEAN*E(IX)
C
C   'Generate Y:
      DO 2 IY = 1, NY
  2   DXY(NX+IY) = YMEAN*E(NX+IY)
C
      RETURN
      END
C
C
C
C
      SUBROUTINE UNIFRM ( ISEED,NX,NY,XMEAN,YMEAN,DXY )
C
C   'Loads DXY with uniformly distributed X,Y.
C
      DIMENSION DXY(NX+NY),U(30)
      NGEN = NX + NY
      CALL SRND ( ISEED,U,NGEN,2,0 )
C
C   'Generate X:
      DO 1 IX = 1, NX
  1   DXY(IX) = U(IX) + XMEAN - 0.5
C
C   'Generate Y:
      DO 2 IY = 1, NY
  2   DXY(NX+IY) = U(NX+IY) + YMEAN - 0.5
C
      RETURN
      END
```

```
C
C
C
C
      SUBROUTINE INORM ( Q,X,IERR )
C
C  'This routine computes the inverse of the normal probability function
C   using a modification of the formula given in Approximations for
C   Digital Computers,  C. Hastings, 1955.
C   The modification consists of the addition of a sinusoidal error
C   reduction term effective in the probability range  10-9 < Q < .5.
C
      REAL*8 Q,X,N,Y,T,B
C
C   'Is Q in the range  0 < Q <= 0.5  ?
C
      IF ( Q.LE. 0.0D0 .OR. Q.GT. 0.5D0 ) THEN
         IERR = 1
         GO TO 10
      END IF
C
      N = DSQRT( -2.0D0 * DLOG(Q) )
      Y = 1.085085260D0 / DSQRT(N)
      T = 2.515517D0 + 8.02853D-1 * N  + 1.0328D-2 * N * N
      B = 1.0D0 + 1.432788D0 * N  + 1.89269D-1*N*N + 1.308D-3*N*N*N
      X = (T/B) - N + 4.434009D-4 * DSIN( 1.1493099D1*Y - 5.789591D0 )
      IERR = 0
   10 RETURN
      END
C
C
C
C
      SUBROUTINE TTEST ( DXY,NX,NY,TVAL )
C
C   'This routine performs a standard 2-sample t-test for differences
C    in the means of X and Y samples.
C
      DIMENSION DXY(NX+NY)
C
C   'Sum X's and X*2's:
      SUMX = 0.
      SUMX2 = 0.
      DO 1 I= 1, NX
      XOB = DXY(I)
      SUMX = SUMX + XOB
   1  SUMX2 = SUMX2 + XOB*XOB
C
C   'Same for Y's:
      SUMY = 0.
      SUMY2 = 0.
      DO 2 J=1, NY
      YOB = DXY(NX+J)
      SUMY = SUMY+ YOB
   2  SUMY2 = SUMY2 + YOB*YOB
C
      DF = REAL( NX + NY - 2 )
      S2P = ( SUMX2 + SUMY2 -(SUMX*SUMX/NX)-(SUMY*SUMY/NY) ) / DF
      T = ( (SUMX/NX)-(SUMY/NY) )/SQRT( S2P*( (1.0/NX) + (1.0/NY) ) )
      TA = ABS(T)
      CALL MDTD ( TA,DF,Q,IER )
      IF ( IER.NE.0 ) PRINT *,'Error in subroutine MDTD'
      TVAL = Q
C
      RETURN
      END
```

LIST OF REFERENCES

1. Fisher, R.A., *The Design of Experiments*, Oliver and Boyd, 1966.

2. Box, G.E.P., Hunter, W.G., and Hunter, J.S., *Statistics for Experimenters - An Introduction to Design, Data Analysis, and Model Building*, John Wiley & Sons, New York, N.Y., 1978.

3. Pitman, E.J.G., Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, *4*, (1937), 119-130.

4. Conover, W.J., *Practical Nonparametric Statistics, 2ed.* John Wiley & Sons, New York, N.Y., 1980.

5. Oden, A. and Wedel, H., Arguments for Fisher's Permutation Test. *The Annals of Statistics*, (1975), Vol. 3, No. 2, 518-520.

6. Lehmann, E.L., *Testing Statistical Hypotheses*, John Wiley & Sons, New York, N.Y., 1959.

7. Richards, F.R., OA 3103 Class Notes, Naval Postgraduate School, Monterey, CA., 1986.

8. Soms, Andrew P., An Algorithm for the Discrete Fisher's Permutation Test. *Journal of the American Statistical Association*, (1977), Vol.72, No.359, 662-664.

9. Garey, M.R. and Johnson, D.S., *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, N.Y., 1979.

10. Lewis, H.R. and Papadimitriou, C.H., The Efficiency of Algorithms. *Scientific American*, Jan. 1978, 96-109.

11. Edgington, Eugene S., *Randomization Tests* (*Second Edition*), Marcel Dekker, Inc., New York and Basel, 1987.

12. Boyett, James M. and Shuster, J.J., Nonparametric One-Sided Tests in Multivariate Analysis with Medical Applications. *Journal of the American Statistical Association*, September 1977, Vol. 72, No. 359, Theory and Methods Section.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chief of Naval Operations (OP 098) Navy Department Washington, D.C. 20350	2
4.	Professor F. Russell Richards Evaluation Technology Incorporated 2150 Garden Rd., Suite B3 Monterey, CA 93940-5327	1
5.	Professor Harold M. Fredricksen Chairman, Dept. of Mathematics, Code 53 Naval Postgraduate School Monterey, CA 93943	1
6.	LT Derek H. Hesse 37 Gorton St. New London, CT 06320	3

END
Feb.
1988
DTIC