AFIT/GSO/ENS/ENC/93D-12

SOLVING THE RANKING AND SELECTION INDIFFERENCE-ZONE FORMULATION FOR NORMAL DISTRIBUTIONS USING COMPUTER SOFTWARE

THESIS

Catherine A. Poston
Captain, USAF

AFIT/GSO/ENS/ENC/93D-12

AFIT/GSO/ENS/ENC/93D-12

SOLVING THE RANKING AND SELECTION INDIFFERENCE-ZONE FORMULATION FOR NORMAL DISTRIBUTIONS USING COMPUTER SOFTWARE

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institu' : of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Space Operations

Accesion For	
NTIS CRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Dist. ibution /	
Availability Codes	
Dist	Avail a d / or Special
A-1	

Catherine A. Poston, B.Sci., Mechanical Engineering
Captain, USAF

15 December, 1993

Approved for public release; distribution unlimited

Thesis Approval

Student: Captain Catherine Poston, USAF

Class: GSO–93D

Thesis Title: SOLVING THE RANKING AND SELECTION INDIFFERENCE-ZONE FORMULATION FOR NORMAL DISTRIBUTIONS USING COMPUTER SOFTWARE

Defense Date: November 19, 1993

Committee:	*Name/Title/Department*	*Signature*
Co-Advisor:	PAUL F. AUCLAIR, Lt Col, USAF Assistant Professor of Operations Research Department of Operational Sciences School of Engineering	Paul F. Auclair
Co-Advisor:	DAVID R. BARR, PhD Associate Professor of Mathematics and Statistics Department of Mathematics and Statistics School of Engineering	David R. Barr

Acknowledgements

I thank my thesis committee, Dr. David Barr and Lt Col Paul Auclair, for their vision, direction, and advice — I could not have had a better balanced advisory team! I am in gratitude to both for being consistently available to review my ideas and answer my anxiety-ridden questions. A special thanks goes to Dr. Barr for providing the theoretical foundation and initial idea for this project and Lt Col Auclair for "smoothing" my writing style. It has been both a joy and a learning experience while under their apprenticeship.

Also, I thank Captain Laura Suzuki for answering all of my *Mathematica* questions (there really were not that many!) and Lt Col Kelso for the numerous discussions on space operation applications.

Many thanks are extended to my fellow GSO-93D classmates and friends who have acted as informal readers and sounding boards. Thanks for all the "insightful" comments and words of encouragement — you guys have kept me smiling throughout the eighteen months!

Finally, I thank my parents whose love and support have given me the confidence to constantly challenge myself and pursue greater goals.

Catherine A. Poston

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table		Page

AFIT/GSO/ENS/ENC/93D-12

Abstract

Ranking and selection procedures are statistical methods used to compare and choose the best among a group of similar statistically distributed populations. The two predominant approaches to solving ranking and selection problems are Gupta's subset selection formulation and Bechhofer's indifference-zone formulation. For the indifference-zone formulation where the populations have equal sample sizes, Barr and Rizvi developed an integral expression of the probability of correct selection (PCS). Given appropriate parameters, the integral expression can be solved to determine the common sample size required to attain a desired PCS.

Tables with selected solutions to the integral expression are available for a variety of population distributions. These tables, however, are not included in any single reference, sometimes require interpolation, and only provide approximate results for the case of unequal sample sizes. Using a computer software program to solve the integral expression for the unknown parameters can eliminate these burdens.

This paper describes the computer software developed to solve the integral expression of the indifference-zone formulation for normally distributed populations having either equal or unequal sample sizes. The software was written in QuickBASIC and *Mathematica*. The QuickBASIC code is a menu-driven interface that develops input files for *Mathematica*. *Mathematica* is the mathematical software package which performs the computationally intensive calculations required to solve the integral expressions. Although limited to comparisons of normal populations, the program can be easily modified to accommodate other distributions. It is hoped that other ranking and selection problems will eventually apply this easy-to-use interface, making the ranking and selection procedure a more common tool in statistical decision making.

SOLVING THE RANKING AND SELECTION INDIFFERENCE-ZONE FORMULATION FOR NORMAL DISTRIBUTIONS USING COMPUTER SOFTWARE

I. Introduction

Occasionally decision makers must select the best of several alternatives. Many decisions have a high impact on society, such as testing for the most effective drugs, choosing the best location for a manufacturing plant, or predicting the winning candidate in a political campaign. The United States Air Force daily makes decisions which impact the defense of the country. These decisions could include choosing a type of aircraft to send on a particular mission, selecting the most effective space-based weapon, or selecting the most reliable or survivable components for the design of a satellite. In all cases, the goal is to select one or more of the k items being considered, based on some factor (usually quantitative) bearing on the decision. These type of problems are classified as ranking and selection problems.

Ranking and selection procedures are statistical methods used to compare and choose the best among a group of similar statistically distributed populations. Barr and Rizvi developed an integral expression that defines the probability of a correct selection (PCS) for the case of populations having equal sample sizes [1:640-646]. Given the number of populations to compare, the number of populations to select, the distribution and parameters of the populations, and the desired PCS; an experimenter can apply the integral expression to solve for the common number of sample observations, n, from each population, that guarantees the stipulated PCS. The experiment is then conducted, collecting n observations from each population.

In many experimental situations, factors such as high costs, risk, scarce resources, or experimental error may result in samples of unequal size. In such an event, the experimenter may be interested in the effect that unequal sample sizes have on the PCS. Barr and Rizvi's integral expression for equal samples does not apply; however, its development is the basis for an unequal sample size formulation.

Since the governing integral expression may be too complicated to solve by hand, published tables provide various combinations of the common sample size, n, and the PCS for a variety of distributions. However, using these tables presents three problems:

- the tables are not included in any single, readily available reference,
- interpolation might be required to determine the appropriate sample size or PCS value, and
- the tables do not provide results for the unequal sample size case.

Prior to this research, a computer program which directly and conveniently solves the appropriate integral expression for the desired variable (*e.g.* n or PCS) had not been developed.

1.1 Research Objective

The purpose of this thesis is to develop a computer program to offer an alternative to the published ranking and selection tables for the normal distribution.

1.2 Scope

Only single-stage procedures are investigated since all cases presented in this thesis effort involve populations having a common and known variance. In a single-stage procedure, one batch of observations is taken as a representative from each population. For populations with unknown variances, at least one stage is used to estimate the variance; the final stage identifies those populations selected as best.

1.3 Thesis Overview

The thesis consists of five chapters. Chapter 2 defines the terminology of the ranking and selection problem and derives the integral expressions for both equal and unequal sample-sized populations. It also applies the resulting expressions to the normal distribution, presents a numerical example, and provides a short summary of specific ranking and selection problems.

Chapter 3 describes the mathematical software and the computer tools chosen to solve the integral expression. The chapter also explains in detail the menu-driven program developed as an interface to the mathematical software package.

Chapter 4 presents a numerical analysis of the values obtained from the mathematical software for normally distributed populations with unequal sample sizes. Based on the analysis, several conjectures are made and a few approaches at proving these conjectures are attempted.

Finally, conclusions are summarized, improvements are recommended and future research options are discussed in Chapter 5.

II. Background

Nearly all procedures used to solve ranking and selection problems are based on either (1) the indifference-zone formulation or (2) the subset-selection formulation [4:296-301]. Although developed in the 1950's, these formulations still provide the basis for current research in ranking and selection procedures. This thesis focuses on the indifference-zone formulation for normally distributed populations.

After some statistical terms are defined, the general indifference-zone integral expression is developed. This integral expression is applied to normally distributed populations with a common, known variance for the cases of equal and unequal sample sizes. The resulting equations are the theoretical foundation for the computer software program developed in Chapter 3. To illustrate their use, a numerical example of the equal sample size case is provided. Finally, a few ranking and selection problems and their associated procedures are discussed to provide a broader context for this research.

2.1 Definition of Terms

The derivation of the indifference-zone integral expression requires the use of several terms, which are defined below.

<u>Population</u> - a large group of data which follows some statistically distributed form [6:2]. Populations are compared in the ranking and selection problem. In this thesis, all populations compared will have the same distributional form. This avoids further complication of the ranking and selection problem. Let k denote the number of populations.

<u>Sample</u> - a subset of a population [6:2]. For populations with equal sample sizes, a sample of size n is considered from each population. For samples of unequal size, the k populations have samples of size n_i $(i = 1, \ldots, k)$.

Parameters - numerical descriptive measures of the population [6:86]. Sufficient statistics are statistics which provide the minimum information needed to describe the population distribution.

Parameter Space - a set consisting of each population's descriptive measure(s) of interest. For instance, if we compare three populations and wish to select the population with the largest mean, then the parameter space will include the mean for each population.

Test Statistic - a function of the sample measurements which acts as an estimate of a population parameter (*e.g.* sample mean, sample variance) [6:429].

Ranking and Selection Procedure - an algorithm that orders the populations according to their "bestness". "Bestness" is determined by the experimental goal and could be the smallest or largest parameter value, the most or least successful population, etc. [4:2]. Ranking and selection procedures are generally classified as:

- Single-Stage - One batch of samples from each population is collected to determine the necessary population information to rank the populations.
- Multi-Stage - A common sample from each population is taken in a first stage to determine an average estimate of the population variance. Other stages provide more samples and better estimates. A final stage uses this estimate to identify those populations selected as best [4:92-95].
- Sequential - After every single sample observation from each observation, the procedure evaluates an estimate of the population variance. The procedure ends according to a predetermined stopping rule that is determined by the experimenter. A selection is made based on the ranking of populations at this stopping point [4:61-87,92-95].

Best selection - based on the rankings, the population or subset of populations with the best ranking parameter(s). Let t be the number of populations to be selected as best.

Probability of Correct Selection (PCS) - the likelihood that the population or set of populations selected is best [4:13- 14].

Indifference Zone (IZ) - a region of the parameter space that defines an insignificant difference between two or more parameters. The experimenter is indifferent between selections whose parameters fall in this zone. [4:5].

Preference Zone (PF) - the complement of the IZ. The region of the parameter space that has a significant difference between two or more parameters. The experimenter has a preference for making a correct selection between selections whose parameters fall in this region. [4:5,9-10].

Indifference Parameter (or Practical Difference, δ) - the significant distance measure between parameters which is generally expressed as a difference or a ratio. Defines the minimum distance required in the PZ, and provides outer bounds for the IZ. The indifference parameter is a numerical value determined prior to experimentation since its value depends on the experimental goal.

Subscript Notation - The ranking and selection literature employs different subscript notations with subtle differences that can mislead the uninitiated. These subscripts can apply to a random variable (*e.g.* Y or X), a parameter (*e.g.* θ), a sample size (*e.g.* n), or a distribution (*e.g.* F_Y).

- bracketed subscripts (*e.g.* [i], [j], [a], [b]) - indicate order. For instance, population parameters are ordered as $\theta_{[1]} \leq \theta_{[2]} \leq \ldots \leq \theta_{[k]}$.

- parenthesized subscripts (*e.g.* (i), (j), (a), (b)) - indicate association with a specific ordered parameter ($Y_{(i)}$ is a random variable associated with the population having an ordered parameter $\theta_{[i]}$).

- unbracketed subscripts (*e.g.* i, j) - indicate neither order nor association with any specific ordered parameter.

2.2 Indifference-Zone Formulation (Integral Development)

The indifference-zone formulation results in an integral expression of the PCS for a single-stage ranking and selection procedure. Development of the integral expression relies heavily on the following definition and theorem presented by Barr and Rizvi [1:642]:

Definition: A cumulative distribution function (CDF) $F(y;\theta)$ with a real parameter θ is said to be stochastically increasing in case,

$$\theta < \theta' \Rightarrow F(y;\theta') \leq F(y;\theta) \qquad \textit{for all } y \tag{2.1}$$

where θ is the parameter and $F(y;\theta)$ the CDF of one population, and θ' and $F(y;\theta')$ are the parameter and CDF of a second population [1:640]. This definition is used in the following theorem:

Theorem: Let $Y_1, \ldots, Y_k$ be k independently distributed random variables with continuous stochastically increasing CDF's, $F(y_i;\theta_i), i = 1, \ldots, k$. Let $\theta_{[1]} \leq \theta_{[2]} \leq \ldots \leq \theta_{[k]}$ denote the ordered values of the θ_i, and let $Y_{(i)}$ denote the random variable with parameter $\theta_{[i]}$. Then, for $t < k$,

$$P = Pr[max(Y_{(1)}, \ldots, Y_{(k-t)}) < min(Y_{(k-t+1)}, \ldots, Y_{(k)})] \tag{2.2}$$

is a non increasing function of $\theta_{[1]}, \ldots, \theta_{[k-t]}$ *and a nondecreasing function of* $\theta_{[k-t+1]}, \ldots, \theta_{[k]}$.

The probability, P, is the basis for obtaining the PCS and can be expressed in two forms [1:640]. Although these forms are developed using separate approaches, they share a common notation. For the k populations introduced in the preceding theorem, let the t $(1 \leq t < k)$ best populations constitute the set $S_2 \equiv \{Y_{(k-t+1)}, \ldots, Y_{(k)}\}$. The remaining $t - k$ populations are included in the set $S_1 \equiv \{Y_{(1)}, \ldots, Y_{(k-t)}\}$.

Approach 1: Let $Y_{(i)}, i = 1, \ldots, k-t$, be a random variable which is in S_1. Considering each possible case in which $Y_{(i)}$ is the maximum observed response among the populations in S_1, the probability in Equation (2.2) can be rewritten as

$$P = Pr(\bigcup_{i=1}^{k-t}\{[max(Y_{(1)}, \ldots, Y_{(k-t)}) = Y_{(i)}] \cap [Y_{(i)} < min(Y_{(k-t+1)}, \ldots, Y_{(k)})]\}). \quad (2.3)$$

The intersection sign indicates that both the equality and inequality expressions must hold to satisfy Equation (2.2). The union of events ranging over $i = 1$ to $k-t$ accounts for all possible populations in S_1 that could yield the maximum observed response. Note further that each probability, conditioned on $Y_{(i)}$, is mutually exclusive. Since

$$P(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} P(E_i) \quad (2.4)$$

for any set of mutually exclusive events $E_1, E_2, \ldots$ [7:30], Equation (3) can then be rewritten as

$$P = \sum_{i=1}^{k-t} Pr\{[max(Y_{(1)}, \ldots, Y_{(k-t)}) = Y_{(i)}] \cap [Y_{(i)} < min(Y_{(k-t+1)}, \ldots, Y_{(k)})]\}. \quad (2.5)$$

Conditioning on $Y_{(i)} = y$,

$$P = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} Pr(\{[max(Y_{(1)}, \ldots, Y_{(k-t)}) = Y_{(i)}] \cap [Y_{(i)} < min(Y_{(k-t+1)}, \ldots, Y_{(k)})]\} \mid Y_{(i)} = y) f_{Y_{(i)}}(y) dy \quad (2.6)$$

where $f_{Y_{(i)}}(y)$ is the probability density function (PDF) of the random variable $Y_{(i)}$ [6:290]. Since each population defined in set S_1 is sampled independently of the populations defined in S_2, Equation (2.6) can be expressed as the product below:

$$P = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} Pr[max(Y_{(1)}, \ldots, Y_{(k-t)}) = y] \cdot Pr[y < min(Y_{(k-t+1)}, \ldots, Y_{(k)})] f_{Y_{(i)}}(y) dy \quad (2.7)$$

Simplifying the first probability expression in Equation (2.7) as the product of the CDF's of the $Y_{(i)}$ $(1 \leq i \leq k-t)$ results in

$$\begin{aligned} Pr[max(Y_{(1)}, \ldots, Y_{(k-t)}) = y] &= P(Y_{(1)} \leq y) P(Y_{(2)} \leq y) \ldots P(Y_{(k-t)} \leq y) \\ &= \prod_{\substack{b=1 \\ b \neq i}}^{k-t} F_{Y_{(b)}}. \end{aligned} \quad (2.8)$$

Since we have defined $Y_{(i)}$ as part of S_1, and we conditioned on $Y_{(i)} = y$, we have already included $Y_{(i)}$ in the probability product (*e.g.* conditionally, $P(Y_{(i)} = y) = 1$). Therefore, in the final expression, the ith factor is not explicitly included. The second probability inside the integral expression in Equation (2.7) is similarly expressed as

$$\begin{aligned} Pr[y < min(Y_{(k-t+1)}, \ldots, Y_{(k)})] &= P(Y_{(k-t+1)} > y) P(Y_{(k-t+2)} > y) \ldots P(Y_{(k)} > y) \\ &= \prod_{a=k-t+1}^{k} [1 - F_{Y_{(a)}}(y)]. \end{aligned} \quad (2.9)$$

Substituting the form of the probabilities given in Equations (2.8) and (2.9), Equation (2.7) can be expressed as

$$P = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b \neq i}}^{k-t} F_{Y_{(b)}}(y) \prod_{a=k-t+1}^{k} [1 - F_{Y_{(a)}}(y)] f_{Y_{(i)}}(y) dy. \qquad (2.10)$$

Equation (2.10) represents the most general form of the indifference-zone integral equation. It applies to cases of both equal and unequal population sample sizes. If all population sample sizes are equal, then Equation (2.10) can be rewritten as

$$P = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b \neq i}}^{k-t} F(y; \theta_{[b]}) \prod_{a=k-t+1}^{k} [1 - F(y; \theta_{[a]})] dF(y; \theta_{[i]}) \qquad (2.11)$$

as a reminder that the functions involved are only dependent on y and the ordered parameters θ.

Suppose we take the limit of P as all the parameter values associated with the populations in S_1 approach the largest parameter value associated with S_1. Let this largest parameter be θ which is defined as the upper limit to the preference zone. Similarly, let all the parameter values associated with the populations in S_2 approach the smallest parameter value associated with S_2. Let this smallest parameter be a function of θ, say $\psi(\theta)$, defined as the lower limit to the preference zone. Let Q be the limit of P. We can express Q as

$$Q = \lim_{\substack{\theta_{[1]}, \ldots, \theta_{[k-t]} \to \theta \\ \theta_{[k-t+1]}, \ldots, \theta_{[k]} \to \psi(\theta)}} P \qquad (2.12)$$

or,

$$Q = \lim_{\substack{\theta_{[b]} \to \theta \\ \theta_{[a]} \to \psi(\theta)}} \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b \neq i}}^{k-t} F(y; \theta_{[b]}) \prod_{a=k-t+1}^{k} [1 - F(y; \theta_{[a]})] dF(y; \theta_{[i]}) \qquad (2.13)$$

which results in

$$Q = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b \neq i}}^{k-t} F(y;\theta) \prod_{a=k-t+1}^{k} [1 - F(y;\psi(\theta))] dF(y;\theta). \tag{2.14}$$

This manipulation expresses Q as a function of only one parameter, namely θ. Since there is no longer an order associated with the $(k-t-1)$ CDF's, as expressed by $F(y;\theta)$, the product of the CDF's can be replaced by the limiting CDF raised to the appropriate power. The product of the limiting complementary CDF's, $(1 - F(y;\psi(\theta)))$, can be expressed in a similar fashion. Using these exponential forms results in

$$Q = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} F^{k-t-1}(y;\theta)[1 - F(y;\psi(\theta))]^t dF(y;\theta). \tag{2.15}$$

As the terms in the sum are free of the index of summation i, Equation (2.15) can also be represented as

$$Q = (k-t) \int_{-\infty}^{\infty} F^{k-t-1}(y;\theta)[1 - F(y;\psi(\theta))]^t dF(y;\theta). \tag{2.16}$$

Equation (2.16) is one of the equations developed in the Barr and Rizvi paper and specifically applies to the case where populations have equal sample sizes. [1:642].

Approach 2: There is a second equivalent way of representing the PCS for the case of equal sample sizes. Since its derivation is similar to the one just completed, some interim steps are omitted.

Reconsider the probability given in Equation (2.2), and define $Y_{(j)}$ as a random variable that is in S_2. We know that to correctly select the best t populations, $Y_{(j)}$ must be larger than the maximum of the set S_1. The PCS can be rewritten as

$$\begin{aligned} P &= Pr(\textstyle\bigcup_{j=k-t+1}^{k}\{ \quad [min(Y_{(k-t+1)}, \ldots, Y_{(k)}) = Y_{(j)}] \cap \\ &\qquad [Y_{(j)} > max(Y_{(1)}, \ldots, Y_{(k-t)})]\}). \end{aligned} \tag{2.17}$$

Using the same mutually exclusive, conditioning, and independence arguments and making substitutions similar to those in the first approach, Equation (2.17) becomes

$$P = \sum_{j=k-t+1}^{k} \int_{-\infty}^{\infty} \prod_{b=1}^{k-t} F_{Y_{(b)}}(y) \prod_{\substack{a=k-t+1 \\ a \neq j}}^{k} [1 - F_{Y_{(a)}}(y)] f_{Y_{(j)}}(y) dy. \tag{2.18}$$

If sample sizes are equal, Equation (2.18) can be rewritten as

$$P = \sum_{j=k-t+1}^{k} \int_{-\infty}^{\infty} \prod_{b=1}^{k-t} F(y;\theta_{[b]}) \prod_{\substack{a=k-t+1 \\ a \neq j}}^{k} [1 - F(y;\theta_{[a]})] dF(y;\theta_{[j]}), \tag{2.19}$$

which is a function of y and the ordered parameters $\theta_{[i]}$, where $1 \leq i \leq k$. Taking the limit of P as $\theta_{[b]} \to \theta$ and $\theta_{[a]} \to \psi(\theta)$, Equation (2.19) becomes

$$Q = \sum_{j=k-t+1}^{k} \int_{-\infty}^{\infty} \prod_{b=1}^{k-t} F(y;\theta) \prod_{\substack{a=k-t+1 \\ a \neq j}}^{k} [1 - F(y;\psi(\theta))] dF(y;\psi(\theta)). \tag{2.20}$$

Similar to approach 1, the product symbols are replaced by corresponding powers to achieve a final expression of

$$Q = t \int_{-\infty}^{\infty} F^{k-t}(y;\theta)[1 - F(y;\psi(\theta))]^{t-1} dF(y;\psi(\theta)). \tag{2.21}$$

Equation (2.21) is the second equivalent integral expression given in the Barr and Rizvi paper and applies to cases of equal sample size populations [1:642].

Both integral expressions (Equations (2.16) and (2.21)) provide the PCS when

- comparing k populations having the same distributional form,
- selecting t populations among them, and
- knowing the indifference parameter (in either the form of $\psi(\theta) - \theta$, or $\psi(\theta)/\theta$).

Typically, an experimenter is interested in the necessary sample size from each population that will guarantee a prescribed PCS. Although it is not apparent from the final expressions, the CDF (the function F) and the derivative of the CDF (the PDF or the function dF) are functions of the common sample size, n. A specific application of Equation (2.21) can best demonstrate this dependence on n.

2.3 Normal Distribution Formulation: Common Known Variance and Equal Sample Size

In this section, Equation (2.21) is used to derive the PCS expression for normally distributed populations of equal sample size and a common, known variance. The resulting equation can then be simplified and implemented into a computer software program which handles the computationally intensive calculation.

For normally distributed populations with a common known variance, the CDF's displayed in Equation (2.21) are readily developed, as shown in Equations (2.22) - (2.24).

$$F(y;\theta) = \Phi[(y-\theta)\sqrt{n}/\sigma] \tag{2.22}$$

$$F(y;\psi(\theta)) = \Phi[(y-\psi(\theta))\sqrt{n}/\sigma] \tag{2.23}$$

$$dF(y;\psi(\theta)) = (\sqrt{n}/\sigma)\varphi[(y-\psi(\theta))\sqrt{n}/\sigma]dy \tag{2.24}$$

where Φ is the CDF and φ is the pdf of the normal distribution (unit 0). For a common sample size, n, the above equations are substituted into Equation (2.21) becoming

$$Q = t\int_{-\infty}^{\infty} \Phi^{k-t}[(y-\theta)\sqrt{n}/\sigma][1-\Phi[(y-\psi(\theta))\sqrt{n}/\sigma]]^{t-1} \cdot$$
$$(\sqrt{n}/\sigma)\varphi[(y-\psi(\theta))\sqrt{n}/\sigma]dy. \tag{2.25}$$

To further simplify the above equation, let

$$u = \Phi[(y-\psi(\theta))\sqrt{n}/\sigma] \tag{2.26}$$

so that

$$du = (\sqrt{n}/\sigma)\varphi[(y-\psi(\theta))\sqrt{n}/\sigma]dy \tag{2.27}$$

and

$$y = \Phi^{-1}(u)\sigma/\sqrt{n} + \psi(\theta). \tag{2.28}$$

Substituting these expressions into Equation (2.25) and transforming the limits of integration,

$$Q = t\int_0^1 \Phi^{k-t}[(\Phi^{-1}(u)\sigma/\sqrt{n} + \psi(\theta) - \theta)(\sqrt{n}/\sigma)](1-u)^{t-1}du. \tag{2.29}$$

If we define the indifference parameter, δ, as

$$\delta = \psi(\theta) - \theta, \tag{2.30}$$

the expression is further simplified to

$$Q = t\int_0^1 \Phi^{k-t}[\Phi^{-1}(u) + \sqrt{n}\delta/\sigma](1-u)^{t-1}du. \tag{2.31}$$

Equation (2.31) is typically used to solve for the PCS, represented as Q, given values for k, t, n, σ, and δ.

2.4 *Normal Distribution Formulation: Common Known Variance, Unequal Sample Sizes*

The indifference-zone integral expressions previously developed are now extended for the case of unequal sample size populations and similarly simplified for computer implementation. Either Equations (2.10) or (2.18) can act as the starting point for this development. To be consistent with the equal sample size formulation for the normal distribution, we begin with Equation (2.18). Equation (2.18) represents the most general case where the goal is to select the best t of k populations. It is repeated here for convenience:

$$P = \sum_{j=k-t+1}^{k} \int_{-\infty}^{\infty} \prod_{b=1}^{k-t} F_{Y_{(b)}}(y) \prod_{\substack{a=k-t+1 \\ a \neq j}}^{k} [1 - F_{Y_{(a)}}(y)] f_{Y_{(j)}}(y) dy. \quad (2.32)$$

Since it is assumed that each population has a unique sample size, the CDF's are not the same. The following substitutions are made into the above equation:

$$F_{Y_{(b)}}(y) = \Phi[(y - \theta_{[b]})\sqrt{n_{(b)}}/\sigma], \quad (2.33)$$

$$F_{Y_{(a)}}(y) = \Phi[(y - \theta_{[a]})\sqrt{n_{(a)}}/\sigma], \quad (2.34)$$

$$F_{Y_{(j)}}(y) = \Phi[(y - \theta_{[j]})\sqrt{n_{(j)}}/\sigma], \quad (2.35)$$

and

$$f_{Y_{(j)}}(y) = (\sqrt{n_{(j)}}/\sigma)\varphi[(y - \theta_{[j]})\sqrt{n_{(j)}}/\sigma]dy. \quad (2.36)$$

Equation (2.32) then becomes

$$
\begin{aligned}
P = & \sum_{j=k-t+1}^{k} \int_{-\infty}^{\infty} \prod_{b=1}^{k-t} \Phi[(y-\theta_{[b]})\sqrt{n_{(b)}}/\sigma] \cdot \\
& \prod_{\substack{a=k-t+1 \\ a \neq j}}^{k} [1-\Phi[(y-\theta_{[a]})\sqrt{n_{(a)}}/\sigma]](\sqrt{n_{(j)}}/\sigma)\varphi[(y-\theta_{[j]})\sqrt{n_{(j)}}/\sigma]dy. \quad (2.37)
\end{aligned}
$$

A change of variables from y to u is made, with the following definitions:

$$u = \Phi[(y-\psi(\theta_{[j]}))\sqrt{n_{(j)}}/\sigma], \tag{2.38}$$

$$du = (\sqrt{n_{(j)}}/\sigma)\varphi[(y-\psi(\theta_{[j]}))\sqrt{n_{(j)}}/\sigma]dy, \tag{2.39}$$

and

$$y = \Phi^{-1}(u)\sigma/\sqrt{n_{(j)}} + \psi(\theta_{[j]}). \tag{2.40}$$

Making these substitutions, Equation (2.37) becomes

$$
\begin{aligned}
P = & \sum_{j=k-t+1}^{k} \int_{0}^{1} \prod_{b=1}^{k-t} \Phi[\sqrt{n_{(b)}/n_{(j)}}\Phi^{-1}(u) + (\theta_{[j]}-\theta_{[b]})\sqrt{n_{(b)}}/\sigma] \cdot \\
& \prod_{\substack{a=k-t+1 \\ a \neq j}}^{k} [1-\Phi[\sqrt{n_{(a)}/n_{(j)}}\Phi^{-1}(u) + (\theta_{[j]}-\theta_{[a]})\sqrt{n_{(a)}}/\sigma]]du. \quad (2.41)
\end{aligned}
$$

If we let the first $k-t$ ordered parameters approach θ, and the last t parameters approach $\psi(\theta)$, the limit of P is given as

$$Q = \lim_{\substack{\theta_{[1]},\ldots,\theta_{[k-t]} \to \theta \\ \theta_{[k-t+1]},\ldots,\theta_{[k]} \to \psi(\theta)}} P$$

$$= \lim_{\substack{\theta_{[b]}\to\theta \\ \theta_{[a]},\theta_{[j]}\to\psi(\theta)}} P, \tag{2.42}$$

which simplifies Equation (2.41) to

$$\begin{aligned} Q &= \sum_{j=k-t+1}^{k} \int_0^1 \prod_{b=1}^{k-t} \Phi[\sqrt{n_{(b)}/n_{(j)}}\Phi^{-1}(u) + (\psi(\theta)-\theta)\sqrt{n_{(b)}}/\sigma] \cdot \\ &\prod_{\substack{a=k-t+1 \\ a\neq j}}^{k} [1 - \Phi[\sqrt{n_{(a)}/n_{(j)}}\Phi^{-1}(u) + (\psi(\theta)-\psi(\theta))\sqrt{n_{(a)}}/\sigma]]du. \end{aligned} \tag{2.43}$$

Substituting the indifference parameter, defined as

$$\delta = \psi(\theta) - \theta, \tag{2.44}$$

into Equation (2.43) yields

$$\begin{aligned} Q &= \textstyle\sum_{j=k-t+1}^{k} \int_0^1 \displaystyle\prod_{b=1}^{k-t} \Phi[\sqrt{n_{(b)}/n_{(j)}}\Phi^{-1}(u) + \delta\sqrt{n_{(b)}}/\sigma] \cdot \\ &\prod_{\substack{a=k-t+1 \\ a\neq j}}^{k} [1 - \Phi[\sqrt{n_{(a)}/n_{(j)}}\Phi^{-1}(u)]]du. \end{aligned} \tag{2.45}$$

Equation (2.45) expresses a form which can be implemented in a computer software program.

In some situations, the experimenter may either not have control over the number of samples provided by each population (due to missing data) or might prefer unequal sample sizes (due to limited resources). Given the observed unequal sample sizes, Equation (2.45) can solve for either PCS or δ. To analyze the effects of unequal sample sizes for normally distributed populations, it helps to further simplify Equation (2.45) into expressions for special cases. For instance, a common goal is to select the one best population among k populations. For this case, t=1 is substituted into Equation (2.45) to become

$$Q = \sum_{j=k}^{k} \int_0^1 \prod_{b=1}^{k-1} \Phi[\sqrt{n_{(b)}/n_{(j)}}\Phi^{-1}(u) + \delta\sqrt{n_{(b)}}/\sigma] \cdot \prod_{\substack{a=k \\ a\neq j}}^{k} [1 - \Phi[\sqrt{n_{(a)}/n_{(j)}}\Phi^{-1}(u)]]du, \quad (2.46)$$

which simplifies to

$$Q = \int_0^1 \prod_{b=1}^{k-1} \Phi[\sqrt{n_{(b)}/n_{(k)}}\Phi^{-1}(u) + \delta\sqrt{n_{(b)}}/\sigma]du. \quad (2.47)$$

A specific case of Equation (2.47) is selecting the one best population from a total of two populations. Substituting k=2 into Equation (2.47),

$$Q = \int_0^1 \prod_{b=1}^{1} \Phi[\sqrt{n_{(b)}/n_{(2)}}\Phi^{-1}(u) + \delta\sqrt{n_{(b)}}/\sigma]du \quad (2.48)$$

which results in

$$Q = \int_0^1 \Phi[\sqrt{n_{(1)}/n_{(2)}}\Phi^{-1}(u) + \delta\sqrt{n_{(1)}}/\sigma]du. \quad (2.49)$$

Equations (2.47) and (2.49) provide formulae that are easier to implement for computer calculation.

2.5 A Numerical Example

A greatly simplified example illustrates the use of the indifference-zone formulae in solving a specific ranking and selection problem.

Consider a satellite system, in early design development, that will have an anti-satellite (ASAT) mission. The satellite system's payload will consist of a kinetic energy weapon whose effectiveness is measured by its ability to hit the geometric center of its target. Suppose that there are three competing contract bids for the mission payload and only one will be selected based on its effectiveness. Suppose further that the kinetic energy weapon accuracy can only be estimated through computer simulation.

An expert is asked to simulate the effectiveness of each weapon system and recommend the most effective one for continued development. It is known from historical simulation runs that the accuracy data (measured in a radial error distance from the target) for each of the proposed weapon systems follows a normal distribution.

The expert and the development agency agree on an experimental objective of correctly selecting the best weapon (least error distance) with at least 95 percent confidence. A practical difference of δ =.004 kilometers and a standard deviation σ =.01 (common to all proposed weapons) should be considered. Substituting the values for δ, σ, and the desired PCS into Equation (2.31),

$$Q = .95 = (1)\int_0^1 \Phi^2[\Phi^{-1}(u) + (\sqrt{n}/.01)(.004)]du.$$

Bechhofer's solutions to a similar expression for different values of n, σ, δ, and PCS are documented in a published table [3:30-37]. A small portion of one of his table is reproduced as Table 2.1.

Bechhofer's table expresses values corresponding to $\sqrt{n}\delta/\sigma$ for a given t, k, and desired PCS and is referenced to throughout this thesis. For this example, a PCS of .95 and a k=3 and t=1 corresponds to a value of 2.7101. The common sample size required to attain the desired PCS as shown below

$$2.7101 = \sqrt{n}\delta/\sigma = \sqrt{n}(.004)/(.01),$$

where

$$n = 45.904.$$

Therefore, the number of observations needed to choose the best mission package with at least 95 percent confidence is 46.

Table 2.1 Partial Bechhofer Table Corresponding to Various PCS Values, To Be Used for Designing Experiments Involving k Normal Populations to Decide which t have the Largest (or Smallest) Population Means.

PCS	k=2 t=1	k=3 t=1	k=4 t=1	k=4 t=2	k=5 t=1
.99	3.29	3.1673	3.7970	3.9323	3.9196
.98	2.9045	3.2533	3.4432	3.5893	3.5722
.97	2.6598	3.0232	3.2198	3.3734	3.3529
.96	2.4759	2.8504	3.0522	3.2117	3.1885
.95	2.3262	2.7101	2.9162	3.0808	3.0552

2.6 Selected Indifference-Zone Ranking and Selection Problems

Bechhofer [3:16-39] and Barr and Rizvi [1:640-646] provided the theoretical foundation for single-stage indifference-zone ranking and selection procedures. Numerous procedures have been developed for problems involving various distributions, including the uniform [2:15-31], binomial [4:103-122], multinomial [4:158-178], gamma [4:328-339], and normal [3:16-39] distributions. This research focuses on the indifference-zone development for two specific problems, namely normally distributed populations with common, known variance and either (1) equal or (2) unequal sample sizes.

For normal populations with a common, known variance and equal sample sizes, Barr and Rizvi's indifference-zone integral expression applies and Bechhofer's table can be used. However, for populations of unequal sample sizes, tables are not available to help solve the selection problem. To rectify this deficiency, Gibbons, Olkin, and Sobel suggest a generalized average sample size,

$$n_0 = [(\sqrt{n_{(1)}} + \sqrt{n_{(2)}} + \ldots + \sqrt{n_{(k)}})/k]^2 \qquad (2.50)$$

computed using the square-mean-root formula [4:50-51]. Substituting n_0 in the equal sample size expressions or Bechhofer's table approximates the desired parameter (*e.g.* PCS, δ). When population variances are not known, single-stage procedures do not apply. For any case of unknown or uncommon variances, exact solutions require multi-stage or sequential procedures [4:61-87,92-95].

The indifference-zone formulation integral expression developed by Barr and Rizvi is applicable to many statistical ranking and selection problems. Here, we have specifically applied it to problems involving normally distributed populations with a common, known variance. The formulae developed in this chapter for normally distributed populations with equal and unequal sample sizes are the basis for the interactive computer software program presented in the following chapter.

III. *Computer Software for Specific Ranking and Selection Problems*

Ranking and selection problems involve the selection of one or more alternatives from a group of many. Information about this selection (*e.g.*, the necessary PCS, the number of observations) can be obtained by solving the indifference-zone integral expressions developed in Chapter 2. This chapter presents a computer software package that solves these integral expressions for any variable of the ranking and selection problem. The software is intended to provide an easily accessible, computationally efficient, and accurate method to solve the integral expression for a variety of experimental situations.

A computer software program that solves these integral expressions was created using two software packages:

- *Mathematica*, a commercial mathematical software package used to perform the complex integrations, and
- QuickBASIC, a BASIC dialect, employed to create a menu-driven shell to the *Mathematica* software program.

This chapter briefly describes these commercial software packages and explains how they were applied to the ranking and selection problem. The final section of this chapter presents the QuickBASIC menu choices and explains the options provided to the user.

3.1 Mathematica

Mathematica is a general software system that evaluates mathematical expressions and creates graphical output. This software was chosen to solve the indifference-zone formulation equation for three reasons:

- it is very easy to work with,

- it is capable of handling a wide variety of complex functions, and
- it can be invoked without understanding the *Mathematica* language itself.

Mathematica Version 2 is available on several operating systems. These include the VMS, UNIX, MS-DOS, Microsoft Windows, NeXT, and Apple operating systems [8]. A *Mathematica* version may be accessed from either a text-based (UNIX) or notebook (PC) interface [8:44-46]. The operating procedures for implementing the ranking and selection software will differ depending on the operating system. Explanations throughout this chapter concentrate on the UNIX version as implemented on a SUN 4/75c microcomputer workstation.

Appendix B includes directions for operating the ranking and selection software using the UNIX operating system.

The *Mathematica* language consists of command statements that are either entered after receiving the the *Mathematica* prompt or read from a text file. Throughout this chapter, *Mathematica* statements are featured in bold. To access the *Mathematica* on a UNIX operating system, the command **math** is typed at the UNIX prompt. The screen input prompt

In[1]:=

indicates that the *Mathematica* system is ready to receive the first command statement. Subsequent inputs are preceded by similar, consecutively numbered prompts. When *Mathematica* reads from a text file, each line in the file acts as a *Mathematica* input statement.

Mathematica contains several unique features that make it ideal for computing the integral expression and interacting with other programming languages. These features include a statistics package, a numerical integration function, a root finding function, and a capability of importing files from outside the *Mathematica* system. The functions that describe these features are briefly discussed.

Statistics Package: This is one of the strongest capabilities that *Mathematica* contributes to the indifference-zone integral computation. The statistics package evaluates some common statistical distribution functions, including the cumulative distribution function, the probability density function, and the quantile function (finding the complimentary CDF) for both discrete and continuous distributions [8:111,585-590].

The statistics package is invoked by the *Mathematica* command

```
<< Statistics'ContinuousDistributions'
```

for continuous distributions, or by

```
<< Statistics'DiscreteDistributions'
```

for discrete distributions. *Mathematica* allows easy access to most common continuous distributions, including the normal, gamma, exponential, and uniform. For example, a function defining a cumulative normal distribution with a zero mean and a standard deviation of one can be expressed as

```
F[t_]: = CDF[NormalDistribution[0,1],t].
```

For any given t value, F[t_] can be numerically evaluated. The probability density function can be similarly defined as

```
f[t_]:=PDF[NormalDistribution[0,1],t],
```

and the quantile as

```
G[t_]:=Quantile[NormalDistribution[0,1],t].
```

The ranking and selection computer software program uses all three of these command statements to evaluate the integral expression.

Numerical Integration: The *Mathematica* **NIntegrate** command numerically integrates a defined function. A function equivalent to 2x can be defined as

```
F[x_]: = 2*x.
```

The integration of 2x from zero to one with respect to x is invoked by

```
NIntegrate[F[x], {x,0,1 }].
```

This *Mathematica* command is used to evaluate the indifference-zone integral for the PCS.

Finding the Root of a Complicated Function: The command FindRoot searches for an approximate solution to a given equation. Let Q[n,t,k] define the indifference-zone integral and let pr be the desired PCS. Given the number of populations, k, the number of populations to be selected, t, and the desired PCS, a numerical approximation for n can be evaluated using the *Mathematica* statement

```
FindRoot[Q[n,t,k]==pr, {n,fst,sec}],
```

where fst and sec are numerical parameters. *Mathematica* first initiates a search in the neighborhood of fst. If a root cannot be found around fst, it will conduct a second search with the alternate parameter, sec.

Ability to Import and Export Text Files: *Mathematica* allows the user to create text files outside of *Mathematica* for import into the *Mathematica* system. Once inside the system, the correct command to retrieve a file is

```
<<ifilename.
```

where ifilename is the name assigned to the input file by the user. *Mathematica* responds to each line of the file as an input statement.

If desired, *Mathematica* responses may be exported to an output file with the use of the *Mathematica* command

>>**ofilename,**

where **ofilename** is the name of the output file as assigned by the user. The output file will contain all of the input statements and *Mathematica* results. It can be viewed, edited, or printed once outside of the *Mathematica* system.

The creation of both an input and an output file avoids direct interaction with *Mathematica*. The following UNIX command, typed at a UNIX prompt, requests *Mathematica* to perform the commands contained in the input file and save the corresponding responses in a separate output file;

```
math <ifilename>ofilename
```

A UNIX prompt signifies the computation is complete and the output file is ready to view. If such an input file has been created for a specific computation, a user can enter this single-line UNIX statement without knowing how to create the file or to program in the *Mathematica* language. Specifically, *Mathematica* can execute the computations necessary for the indifference-zone integral calculation without a direct user-*Mathematica* interface.

A menu-driven software program can make *Mathematica* calculations transparent and provide the user with multiple options for solving a variety of ranking and selection problems. QuickBASIC is the language that was chosen to create this menu-driven shell.

3.2 QuickBASIC

QuickBASIC is a dialect of the BASIC language. It offers a user-friendly environment that contains an easy-to-use menu structure, a syntax-checking editor and compiler, 1 and 1 editing capabilities, and full debugging resources. A QuickBASIC program can execute a single line at a time, with the system identifying and reporting errors as the code is being entered [5:1].

In addition, QuickBASIC is easilv accessible. Microsoft QuickBASIC is usually provided as an application to Microsoft Windows. Most computers that have DOS 5.0 also include an interactive version of the QuickBASIC programming language, referred to as QBASIC.

3.3 QuickBASIC Ranking and Selection Computer Shell

The QuickBASIC software shell provides an interface between the user and the *Mathematica* implementation of specific ranking and selection problems. It guides the user through a series of menu-driven options and requests selections based on the goal of the experiment. The choices selected provide the information necessary to create a unique input file containing *Mathematica* commands.

This section describes the menu structure, and several general features of the program. Appendix C contains the QuickBASIC code that created this menu structure.

3.3.1 Structure. The menu-driven program is structured into five levels as shown in Figure 3.1. These menu levels and each menu option are briefly described.

LEVEL I Menu: This level contains one menu:

Ranking and Selection Problem Menu. The menu allows the choice of a normal ranking and selection procedure based on either the means or the variances of the populations. The option to rank populations by their variances has not yet been implemented due to the scope of this thesis. Figure 3.2 displays the Level I Menu Options.

- Normal, Ranking Means. This option is chosen when the k populations being compared follow a normal distribution. Populations are ranked based on each population mean. The population means and the ranking of these means are not known.

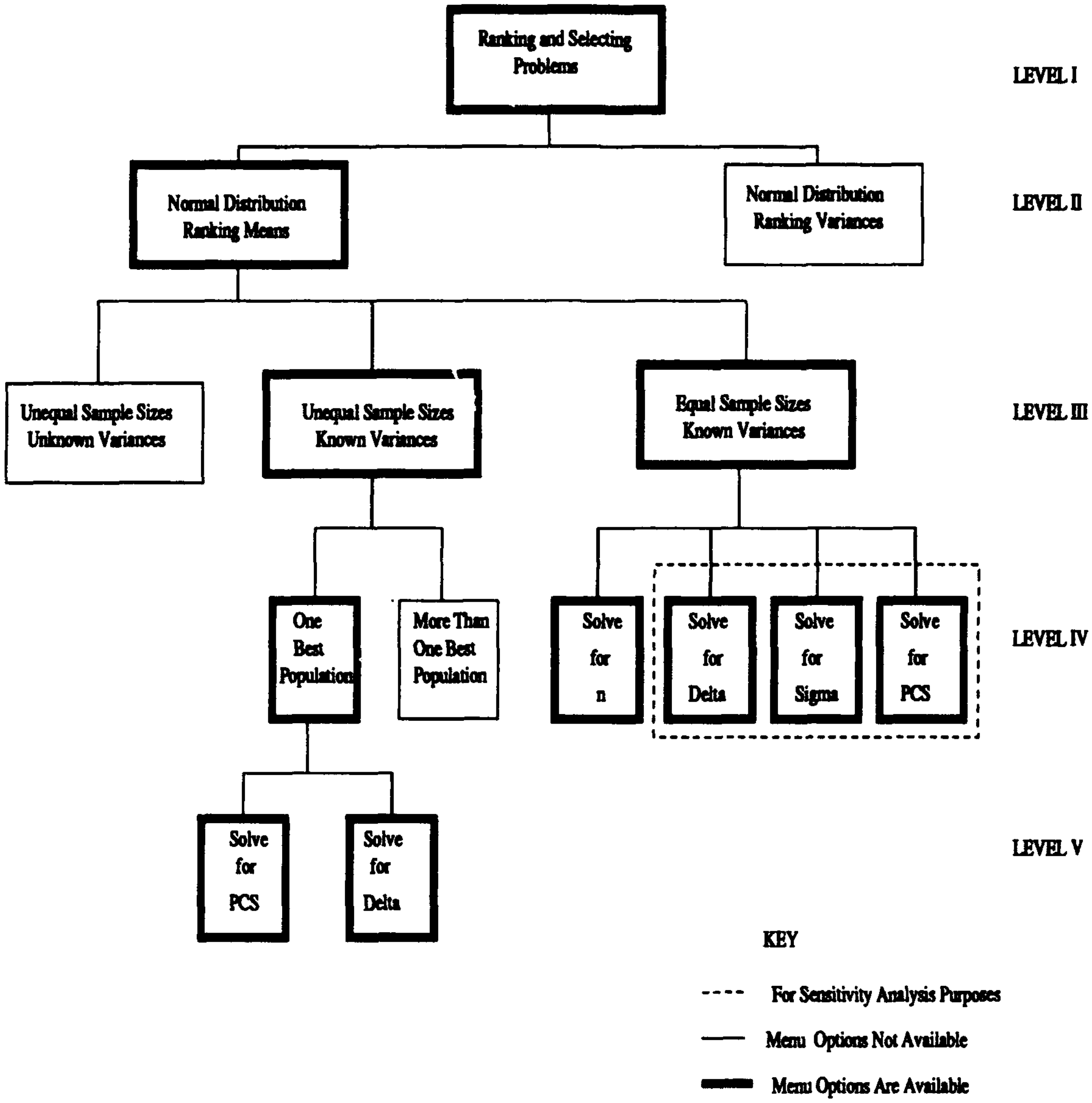

Figure 3.1 QuickBASIC Ranking and Selection Computer Shell Menu Structure

* Normal, Ranking Variances. Although this choice currently is not available, this menu option would be chosen when the normally distributed k populations are ranked based upon each unique population variance.

```
RANKING AND SELECTION PROBLEMS

(Select option.  You do not have to press ENTER)

Options for the Distributions are:

   (1) Normal - Ranking Means
 * (2) Normal - Ranking Variances
   (3) Quit

 (*) indicates option is not available

 Select Option:
```

Figure 3.2 Level I Menu Options: Ranking and Selection Problems

LEVEL II Menu: This level contains two menus:

1. Normal, Ranking Means Menu. The user chooses an option based on whether the population sample sizes are common and whether the population variances are known. The option for procedures involving populations with unequal sample sizes and unknown variances is not implemented due to the scope of the thesis. Options from this menu are displayed in Figure 3.3.

- Equal Sample Sizes, Known Variance. This option is chosen when there is a common and unknown sample size, n, among all k populations. The population variances must be known and equal.
- Unequal Sample Sizes, Known Variance. A user chooses this option when it is impossible to obtain a common sample size from each of the k populations. The number of observations sampled from each population must be known, since the software package does not solve for the most efficient distribution of sample sizes between populations.

```
Normal - Ranking Means Menu
(Select option.  You do not have to press ENTER)
Options are:
  (1) Equal Sample Sizes, Known Variance
  (2) Unequal Sample Sizes, Known Variance
* (3) Unequal Sample Sizes, Unknown Variance
  (4) Quit
(*) indicates option is not available
Select Option:
```

Figure 3.3 Level II Menu Options: Normal, Ranking Means Menu

- Unequal Sample Sizes, Unknown Variance. Although not available, it is included in the menu structure as a possible future addition to the computer program.

2. Normal, Ranking Variance Menu. This menu has yet to be developed but should contain options similar to the ranking means menu.

LEVEL III Menu: This level contains three menus:

1. Normal, Ranking Means, Equal Sample Size, Known Variance Menu. Four options are available as shown in Figure 3.4. The primary goal of the experimenter is to determine the common sample size among all populations to guarantee a PCS. However, options to solve for the PCS, δ, and σ are included for sensitivity analysis purposes.

```
Normal-Ranking Means-
Equal Sample Size, Known Variance Menu
(Select option.  You do not have to press ENTER)
Options are:
    (1) Solve for Common Sample Size, n
    (2) Solve for Probability of Correct Selection,  PCS
    (3) Solve for the Indifference Parameter, Delta
    (4) Solve for Standard Deviation, Sigma
    (5) Quit
Select Option:
```

Figure 3.4 Level III Menu Options: Normal, Ranking Means, Equal Sample Size, Known Variance Menu

- Solve for Common Sample Size, n. The goal is to determine the number of observations required from each population to guarantee a desired PCS.
- Solve for Probability of Correct Selection, PCS. The goal is to determine the probability that the best population(s) is (are) correctly selected.
- Solve for the Indifference Parameter, Delta. The goal is to determine the smallest detectable practical difference between the best and next best populations, given the desired PCS and the sample sizes. The practical difference is measured as the difference in population means.

- Solve for Standard Deviation, Sigma. The goal is to determine the standard deviation, common to all k populations, which satisfies the integral expression.

2. Normal, Ranking Means,Unequal Sample Size, Known Variance Menu. This level III menu provides choices dependent on the user's goal for problems involving unequal population samples. Two options are available and are depicted in Figure 3.5.

```
Normal-Ranking Means-
Unequal Sample Sizes, Known Variance
(Select option.  You do not have to press ENTER)
Options are:
  (1) Number of Best Populations is 1
* (2) Number of Best Populations is More Than 1
  (3) Quit
(*) indicates option is not available

Select Option:
```

Figure 3.5 Level III Menu Options: Normal, Ranking Means, Unequal Sample Size, Known Variance Menu

- Number of Best Populations is 1. The goal is to select the best among k populations. The options allow calculation of the probability of correct selection or the indifference parameter associated with this best selection.
- Number of Best Populations Is More Than 1. Although not implemented, this menu option could be chosen when the k populations do not have a

common number of sample observations and the goal is to select two or more best populations.

3. Normal, Ranking Means, Unequal Sample Sizes, Unknown Variance Menu. This menu could include multi-stage or sequential procedure options to estimate a population variance. The development is left for future computer implementation.

LEVEL IV Menus: This level contains six menus; four for cases of equal population samples, and two for cases of unequal population samples.

Common Sample Size Menus. Each menu contains directions for the user to enter numerical values from the keyboard. The QuickBASIC software code formats these values into *Mathematica* language and creates a *Mathematica* input file. This input file invokes one or more of the *Mathematica* computational files listed in Appendix D. These computational files solve the indifference-zone integral expression for the particular parameter requested by the user in the menu selection. The four *Mathematica* input file examples, corresponding to the four menu choices, are depicted in Appendix E.

Three of the four menus (solving for n, σ, or δ) include a search parameter option. In these menus, the program allows users to either submit their own search parameters or to accept computed estimates. Appendix D explains the equations and the computational files used to estimate the search values. The search value option is not available in the 'Solve for PCS' menu since a root finding function is not necessary to determine the PCS.

The four common-sample-size menus that are available to the user include the following;

1. Normal, Ranking Means, Equal Sample Size, Known Variance, Solve for n. Figure 3.6 displays the directions for this option. The numerical values for the

indifference parameter δ, the standard deviation σ, the number of competing populations k, the number of best populations t, and the desired PCS are necessary to calculate an exact value for n.

2. Normal, Ranking Means, Equal Sample Size, Known Variance, Solve for PCS. Figure 3.7 displays the directions for this option. Values for δ, σ, k, t, and n must be provided by the user.

3. Normal, Ranking Means, Equal Sample Size, Known Variance,Solve for Delta. Figure 3.8 displays the directions for this option. Values for σ, n, t, k, and PCS must be provided by the user.

4. Normal, Ranking Means, Equal Sample Size, Known Variance, Solve for Sigma. Figure 3.9 displays the directions for this option. Values for δ, n, k, t, and PCS must be provided by the user.

Unequal Sample Size Menus.

5. One Best Population Menu. The two available options are depicted in Figure 3.10. In testing cases using the *Mathematica* computational algorithm, it was discovered that the PCS and δ values are dependent on how the sample sizes are associated with the ordered populations. As defined in Chapter 2, $n_{(i)}$ is the sample size associated with the population having ordered parameter $\theta_{[i]}$. Since the sample sizes are assigned to the k populations without knowledge of which one is best, there are $\binom{k}{1}$, or k possible associations between the k sample sizes and the best population. The QuickBASIC shell program writes all k computational statements in the *Mathematica* language and *Mathematica* performs k calculations of the PCS or δ to determine the minimum value. Computation time increases as k gets larger.

```
Normal-Ranking Means-
Equal Sample Sizes, Known Variance - Solve for n
(Input value and hit the ENTER key)
    (1) Enter indifference parameter, delta?
    (2) Enter standard deviation, sigma?
    (3) Enter number of populations to be ranked?
    (4) Enter number of best populations desired?
    (5) Enter desired probability of correct
                selection?
                    Do you want to enter your own search
                    values for n (Y or N)?
                    (if 'N' then the program computes values
                     for the search and items (6) and (7) are
                      skipped.) Y
    (6) Enter first search value for n?
    (7) Enter second search value for n?
    (8) Enter the drive you want the exported file
                on either A, B, or C (default is C):
    (9) Enter the name of the data file,
                  (.txt data extension assumed)
                  (default is norm.txt):
```

Figure 3.6 Level IV Menu: Normal, Ranking Means, Equal Sample Size, Known Variance, Solve for n Display

```
Normal-Ranking Means-
Equal Sample Sizes, Known Variance-Solve for PCS
(Input value and hit the ENTER key)
   (1) Enter indifference parameter, delta?
   (2) Enter standard deviation, sigma?
   (3) Enter number of populations to be ranked?
   (4) Enter number of best populations desired?
   (5) Enter sample size?
   (6) Enter the drive you want the exported file
                on either A, B, or C (default is C):
   (7) Enter the name of the data file,
                (.txt data extension assumed):
                (default is norm1):
```

Figure 3.7 Level IV Menu: Normal, Ranking Means, Equal Sample Size, Known Variance, Solve for PCS Display

```
Normal-Ranking Means-
Equal Sample Size, Known Variance-Solve for Delta
(Input value and hit the ENTER key)
   (1) Enter standard deviation, sigma?
   (2) Enter sample size?
   (3) Enter number of populations to be ranked?
   (4) Enter number of best populations desired?
   (5) Enter desired probability of correct
               selection?
                   Do you want to enter your own search
                   values for delta (Y or N)?
                   (If 'N' then the program computes values
                    for the search and items (6) and (7) are
                     skipped.) Y
    (6) Enter first search value for delta?
    (7) Enter second search value for delta?
    (8) Enter the drive you want the exported file
               on either A, B, or C (default is C):
    (9) Enter the name of the data file,
               (.txt data extension assumed),
               (default is norm1):
```

Figure 3.8 Level IV Menu: Normal, Ranking Means, Equal Sample Size, Known Variance, Solve for Delta Display

```
Normal-Ranking Means-
Equal Sample Sizes, Known Variance-Solve for Sigma
(Input value and hit the ENTER key)
   (1) Enter indifference parameter, delta?
   (2) Enter sample size?
   (3) Enter number of populations to be ranked?
   (4) Enter number of best populations desired?
   (5) Enter desired probability of correct
                 selection?
                    Do you want to enter your own search
                    values for sigma (Y or N)?
                     (If 'N' then the program computes values
                      for the search and items (6) and (7) are
                       skipped.) Y
    (6) Enter first search value for sigma?
    (7) Enter second search value for sigma?
    (8) Enter the drive you want the exported file
                 on either A, B, or C (default is C):
    (9) Enter the name of the data file,
                 (.txt data extension assumed),
                 (default is norm1):
```

Figure 3.9 Level IV Menu: Normal-Ranking Means-Equal Sample Size, Known Variance-Solve for Sigma Display

```
Normal-Ranking Means-
Unequal Sample Size, Common Variance
One Best Population
(Select option.  You do not have to press ENTER)
 Options are:
    (1) Solving for PCS When Sample Sizes are Known
    (2) Solving for Delta When Sample Sizes are Known
    (3) Quit
Select Option:
```

Figure 3.10 Level IV Menu Options: Normal, Ranking Means, Unequal Sample Size, Known Variance, One Best Population Menu

- Solving for PCS When Sample Sizes are Known. This option leads to solving the unequal sample size indifference-zone integral expression for the PCS.
- Solving for Delta When Sample Sizes are Known. This menu option leads to solving the unequal sample size indifference-zone integral expression for the indifference parameter.

6. Number of Best Populations is More Than 1. This menu is not developed. The extension of the algorithms developed for the $t = 1$ case required excessive computation time when extended to cases where $t > 1$. Other approaches should be developed to implement this option.

LEVEL V Menus: This level contains two menus for problems involving unequal population samples. Both require numerical values to be supplied by the user.

```
Normal-Ranking Means-
Unequal Sample Sizes, Known Variance
One Best Population-Solve for PCS
(Input value and hit the ENTER key)
    (1) Enter indifference parameter, delta?
    (2) Enter standard deviation, sigma?
    (3) Enter number of populations to be ranked?
    (4) Enter first sample size?
        Enter next sample size?
```

Figure 3.11 Level V Menu: Normal, Ranking Means, Unequal Sample Size, Known Variance, Solve for the PCS Display

1. Normal, Ranking Means, Unequal Sample Size, Known Variance, Solve for PCS Menu. Figure 3.11 displays the directions for this option. The user provides numerical values for δ, σ, and k. The program also requests the sample size for each population. Sample sizes can be placed in any order, since the program permutes the sample size order to produce various PCS's. *Mathematica* produces all k PCS values and determines the minimum.

2. Normal, Ranking Means, Unequal Sample Size, Known Variance, Solve for Delta Menu. Figure 3.12 displays the directions for this option. Numerical values for the PCS, σ, k, and approximate search values for δ must be provided by the user. The user inputs the sample sizes in any order upon request. Since δ is dependent upon the association between the sample sizes and the best population, k permutations of the associations will produce different δ values. *Mathematica* performs k calculations, produces the k values of δ, and determines the minimum δ.

```
Normal-Ranking Means-
Unequal Sample Sizes, Known Variance
1 Best Population-Solve for Delta
(Input value and hit the ENTER key)
   (1) Enter the probability of correct selection, PCS?
   (2) Enter standard deviation, sigma?
   (3) Enter number of populations to be ranked?
   (4) Enter first search value for delta?
   (5) Enter second search value for delta?
   (6) Enter first sample size?
       Enter next sample size?
```

Figure 3.12 Level V Menu: Normal, Ranking Means, Unequal Sample Size, Known Variance, Solving for Delta Display

3.3.2 Features. Several user-friendly features were added to the QuickBASIC computer shell. The features include:

- An option to quit at any menu level (except at Level V).
- The capability to specify the disk drive on which the QuickBASIC file will be saved. The default is the 'C' drive.
- The capability to specify the name of the output file. The default is 'norm1.txt'.
- An option to recover after a menu choice selection or a numerical value input. The program displays the option chosen and questions the user on their selection.
- A display which depicts the values chosen by the user prior to file creation. Users are given a last chance to change any numerical values (only available

on Level IV Equal Sample Size Displays). An example display is depicted in Figure 3.13.

```
The following are the values that you assigned:
   (1) delta= 4
   (2) sigma= 10
   (3) number of populations= 3
   (4) number of best populations= 1
   (5) desired PCS= .95
   (6) first search value= 30
   (7) second search value=35
   (8) drive to store file=C
   (9) name of exported file=norm1
Do you want to make any changes (Y or N or Q to Quit)?
```

Figure 3.13 An Example Value Check Provided by the QuickBASIC Software Program(When the Program Provides Search Values)

Although the current computer program is limited in scope, it demonstrates the feasibility of using *Mathematica* with a QuickBASIC interface to solve the integral expression characteristic of ranking and selection problems. *Mathematica*'s computational capability allows numerical analysis of the indifference-zone ranking and selection procedures. The next chapter demonstrates this analysis for problems involving unequal sample size populations.

IV. Computer Program Implementation and Analysis

Chapter 2 developed the indifference-zone integral equations for both equal and unequal population sample sizes. These equations were simplified to a form easily manipulated by a mathematical computer software program. Chapter 3 explained the computer program implemented to solve the integral equations for a variety of experimental conditions. This chapter analyzes the numerical results obtained from the computer program. The numerical values for both the equal and unequal sample size cases are validated against Bechhofer's table. For the case of unequal sample sizes, computer results are compared to the estimated PCS obtained using the Gibbons, Olkin, and Sobel approximation as presented in Section 2.6. Several conjectures for the unequal sample size case are made based on empirical observations. Although the attempts to prove these conjectures for the most general cases were inconclusive, they are included for completeness.

4.1 Populations with Equal Sample Sizes

Equation (2.31) was developed to solve ranking and selection problems involving normally distributed populations with equal sample sizes. This equation was rewritten in *Mathematica* code, placed in a file for *Mathematica* input, and used to calculate numerical values for n. Appendix D contains this and all other *Mathematica* computational files.

Ten cases were examined to compare the values obtained from the computer software to those listed in Bechhofer's table [3:30-37]. The table includes several hundred cases, where k and PCS span from 2 to 14 and .200 to .999 respectively. Because of the massive amount of cases that could be tested, values of k, t and PCS were arbitrarily chosen from the table. Values of σ and δ remained fixed at 10 and 4, respectively. For all attempted cases, the values of $\sqrt{n}\delta/\sigma$ that were obtained from the computer software program were compared with values obtained from Bechhofer's table. Most values matched exactly out to four decimal places. This result suggests that the *Mathematica* coded formula can be reasonably relied upon to solve the integral expression for the attempted equal sample sizes cases. Cases outside of the tested range should be investigated to further substantiate the validation. Table 4.1 summarizes the specific cases examined.

Table 4.1 A Comparison Between Bechhofer's Table Values and *Mathematica* for Equal Sample Size Populations.

Case	*PCS*	Bechhofer $\sqrt{n}\delta/\sigma$	*Mathematica* $\sqrt{n}\delta/\sigma$
k=2 t=1	0.95	2.3262	2.3261
k=3 t=1	0.88	2.0899	2.0899
k=4 t=1	0.55	0.9936	0.9936
k=4 t=2	0.97	3.3734	3.3733
k=5 t=1	0.90	2.5997	2.5997
k=5 t=2	0.70	1.9765	1.9765
k=6 t=1	0.60	1.4575	1.4575
k=6 t=2	0.99	4.2244	4.2244
k=6 t=3	0.82	2.6535	2.6534
k=7 t=1	0.75	2.0626	2.0626

There is a distinct advantage in using *Mathematica* to determine n or any of the other parameters of interest. The computer program calculates an exact value for a parameter given any PCS while Bechhofer's table lists specific PCS values. Experiments involving PCS values not found in Bechhofer's table require interpolation to use them. The computer program avoids the need to interpolate and provides more precise numerical solutions.

A minor disadvantage was discovered in validating the cases using the computer program. The indifference-zone integral expression using *Mathematica* is computationally intensive and time consuming. Although the program takes between 30 and 60 seconds to calculate the PCS, it takes up to three minutes to solve for n. This can be cumbersome when examining more then one experimental case.

4.2 Populations With Unequal Sample Sizes

4.2.1 Computer Program Validation with Bechhofer's Tables.

Equation (2.45) was developed to solve ranking and selection problems involving populations of unequal sample sizes. A *Mathematica* computational file was created to compute the unequal sample size expression. The file is essentially the backbone of the computer software program for unequal sample size applications and is partially validated in this section.

The same tables used to validate the equal sample size *Mathematica* formulation can also be used to partly validate the unequal sample size computation. To do so, ten cases defined by their k, t, and PCS values were arbitrarily chosen from Bechhofer's table. A common value of n was determined from the table for each case. In the *Mathematica* computation file, all k sample sizes were assigned this same sample size and the PCS was computed. Since the equal sample size problem is a special case of the unequal sample size problem, the *Mathematica* results should agree with Bechhofer's table. A comparison between the PCS values as computed by *Mathematica* and the PCS values derived from Bechhofer's table are depicted in Table 4.2. The PCS values are approximately equal and vary only due to rounding error associated with calculating n from Bechhofer's table.

4.2.2 Computer Program Validation with MATHCAD Software. The previous method validated the unequal sample size *Mathematica* computation for the case of equal sample sizes. However, this method cannot confirm a correct PCS when unequal sample sizes are involved. To validate the *Mathematica* computation, MATHCAD, another mathematical software program, was coded with the same unequal sample size integral expression. Since *Mathematica* has a numerical precision out to 16 decimal places and MATHCAD has a precision out to 15 decimal places, a 0 between PCS values is applicable and meaningful. The MATHCAD code is presented in Appendix F. Five cases for selecting the best of two populations with unequal sample sizes were examined. The computed PCS values are compared in Table 4.3. PCS values are virtually identical in every case, further confirming that *Mathematica* is correctly performing the unequal sample size computation.

Table 4.2 A Comparison of Specific Numeric PCS Values as Obtained from Bechhofer's Table and *Mathematica* for Unequal Sample Size Populations. Note: δ/σ =.4 For All Cases.

Case	n	Bechhofer *PCS*	*Mathematica* *PCS*
k=4 t=2	59.3208	0.950000	0.949997
k=5 t=1	8.5936	0.550000	0.550001
k=5 t=2	34.1960	0.800000	0.800008
k=6 t=1	51.2047	0.920000	0.920007
k=6 t=2	15.3938	0.500000	0.499987
k=6 t=3	35.6617	0.750000	0.749986
k=7 t=1	1.3248	0.250000	0.249996
k=7 t=3	46.4740	0.800000	0.800006

Table 4.3 A Comparison of Specific Numeric PCS Values as Obtained from *Mathematica* and *MATHCAD* Computer Software Programs for Unequal Sample Size Populations in the Case of k=2 and t=1. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	*Mathematica* *PCS*	*MATHCAD* *PCS*
15	20	0.879217	0.879216
30	24	0.927936	0.927936
29	30	0.937732	0.937732
60	45	0.978739	0.978740
57	60	0.984715	0.984717

4.2.3 Computer Program Validation With Published Methods. Gibbons, Olkin and Sobel's (GOS) method for addressing the unequal sample size problem was presented in Section 2.6. Their procedure estimates a common average sample size, n_0, in order to apply Bechhofer's table. A comparison between the PCS values obtained from the unequal sample size *Mathematica* computation and those estimated by the GOS method further validates the computer program calculation.

Eight cases involving populations of unequal sample sizes were examined using the square-mean-root formula (Equation (2.50)) to determine an average sample size, n_0. Using n_0 as the common sample size, the equal sample size *Mathematica* program computed the GOS PCS values listed in Table 4.4. These values were compared to the results of direct *Mathematica* computation of the PCS for the unequal sample size case. The close agreement of the values for each case provides another indication that *Mathematica* is correctly performing the unequal sample size computation.

Table 4.4 A Comparison of Specific Numeric PCS Values as Obtained from *Mathematica* and the Gibbons, Olkin, and Sobel (GOS) Method for Unequal Sample Size Populations. Note: δ/σ =.4 For All Cases.

		Sample Sizes						*Mathematica*	GOS
k	t	$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	n_0	*PCS*	*PCS*
2	1	15	20				17.410254	0.879217	0.881035
2	1	60	45				52.230762	0.978739	0.979530
3	1	50	50	45			48.304073	0.953650	0.955212
3	2	50	50	45			48.304073	0.955703	0.955212
4	1	50	50	48	45		48.227965	0.935356	0.937845
4	2	50	50	48	45		48.227965	0.918771	0.918993
5	2	52	50	48	45	52	49.363776	0.892803	0.893866
5	3	52	50	48	45	52	49.363776	0.894263	0.893866

4.2.4 Empirical Computer Program Findings. For populations of unequal sample sizes, many cases were investigated to determine any pattern that might further simplify computer program implementation. Several conjectures are made based on this analysis.

4.2.4.1 Conjecture 1:. When one sample size is smaller than the rest, randomly assigning the reduced sample size to the various populations produces the same PCS value for the cases of k=2, 4, or 6 and $t = k/2$.

Prior to experimentation, the experimenter does not know the order of the population parameters (*e.g.* which populations have the larger or smaller means). Therefore, the experimenter does not know which sample size is associated with the best or worst population. Equation (2.45) implies that the PCS is dependent on the association between the sample sizes and the ordered populations. This association will henceforth be referred to as the sample size association. A specific sample size association is the one-to-one correspondence of the unordered population sample sizes, n_a, $1 \leq a \leq k$, to each of the populations with ordered parameters, $\theta_{[i]}$, where $\theta_{[1]} \leq \theta_{[i]} \leq \theta_{[k]}$.

When preparing the *Mathematica* input file to calculate either the PCS or δ, the QuickBASIC computer software program allows the user to enter the sample sizes in some initial, arbitrary order. The program associates a sample size to one of the ordered populations based on this initial order (*i.e.* initially, n_1 is assigned to the population with an ordered parameter, $\theta_{[1]}$, n_2 is assigned to the population with a parameter $\theta_{[2]}$, *etc.*) The program then produces *Mathematica* statements to calculate PCS or δ values for this assignment and every other possible sample size association that produces a unique numerical PCS value.

For the particular case of k=2, 4, or 6 and t=$k/2$ when one sample size is unequal, every sample size association appears to produce the same PCS or δ values. This phenomena can be exploited to enhance the efficiency of the computer software program. Instead of computing values for every sample size association, only one calculation is necessary.

This conjecture was based on the empirical results observed while testing the *Mathematica* computation file for the unequal sample size case. The conjecture was tested for

- k =2 and t =1,
- k =4 and t =2, and

- k =6 and t =3.

Various sample sizes were arbitrarily selected for each test condition. For the k populations considered, $k-1$ were assigned the nominal sample size, while the remaining population was allotted a smaller number of sample observations. Throughout the investigation, σ and δ remained fixed at 10 and 4 respectively. The resulting comparisons are presented in Tables 4.5, 4.6, and 4.7. The comparisons demonstrate the independence of the PCS to the sample size associations when only one population is alloted less then the nominal sample size.

Table 4.5 PCS Values as a Result of Varying Sample Size Associations for Unequal Sample Size Populations in the Case of k =2, t =1. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	PCS
15	14	0.859124
14	15	0.859124
15	13	0.854424
13	15	0.854424
100	90	0.997047
90	100	0.997047

Cases in which two of the k populations were allotted one less observation were also examined. PCS results are located in Tables H.1 and H.2 of Appendix H for the cases of k =4, t =2 and k =6, t =3. When having more then one sample size that is less then the nominal sample size, the PCS value varies depending on the sample size association. Therefore, conjecture 1 appears to be true only when there is one unequal sample size.

Conjecture 1 could possibly apply to any even k value when $t=k/2$, but the result could not be proven for the general case. To document the approaches that failed to produce the desired result, we include our several attempts in proving Conjecture 1 using the simplest case of k=2 and t=1.

Table 4.6 PCS Values as a Result of Varying Sample Size Association for Unequal Sample Size Populations in the Case of k =4, t =2. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	PCS
15	15	15	14	0.64207
15	15	14	15	0.64207
15	14	15	15	0.64207
14	15	15	15	0.64207
15	15	15	13	0.63739
15	15	13	15	0.63739
15	13	15	15	0.63739
13	15	15	15	0.63739

Table 4.7 PCS Values as a Result of Varying Sample Size Association for Unequal Sample Size Populations in the Case of k =6, t =3. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	$n_{(6)}$	PCS
15	15	15	15	15	14	0.452412
15	15	15	15	14	15	0.452412
15	15	15	14	15	15	0.452412
15	15	14	15	15	15	0.452412
15	14	15	15	15	15	0.452412
14	15	15	15	15	15	0.452412
15	15	15	15	15	13	0.448719
15	15	15	15	13	15	0.448719
15	15	15	13	15	15	0.448719
15	15	13	15	15	15	0.448719
15	13	15	15	15	15	0.448719
13	15	15	15	15	15	0.448719

Equation (2.49) displays the integral expression of the PCS for the specific case of k=2 and t=1. It is not intuitively obvious from this equation that the PCS is independent of the sample size association. In attempting to prove the conjecture that the same PCS value is produced when randomly assigning the smaller sample size to the various populations, several analytical methods are described below.

1. Equation Substitution using Total Sample Size, N, and a Proportion Variable, λ.

   For the case of k=2 and t= 1, let the first sample size that the user initially enters into the computer program be denoted n_1 and the second sample size be n_2. In Equation (2.49), $n_{(1)}$ denotes the sample size that is associated with the lowest ranked population while $n_{(2)}$ is the sample size associated with the highest ranked population. The first possible sample size association is

   $$n_{(1)} = n_1 \tag{4.1}$$

   and

   $$n_{(2)} = n_2. \tag{4.2}$$

   When substituting this into Equation (2.49), this assignment directly results in the following equation;

   $$Q = \int_0^1 \Phi[\sqrt{n_1/n_2}\Phi^{-1}(u) + \delta\sqrt{n_1}/\sigma]du. \tag{4.3}$$

   The second possible sample size association is

   $$n_{(1)} = n_2 \tag{4.4}$$

   and

   $$n_{(2)} = n_1, \tag{4.5}$$

which results in the following equation;

$$Q = \int_0^1 \Phi[\sqrt{n_2/n_1}\Phi^{-1}(u) + \delta\sqrt{n_2}/\sigma]du. \qquad (4.6)$$

Empirical evidence suggests that Equations (4.3) and (4.6) produce the same PCS, given identical σ, δ, n_1 and n_2. However, it is difficult to show that these equations are equal except for specific PCS values. In order to better demonstrate the equivalence, the total sample size, N, and a variable, λ, which partitions the total sample size into separate samples, are substituted into Equations (4.3) and (4.6).

The total sample size, N, can be defined as

$$N = n_1 + n_2 = n_{(1)} + n_{(2)}. \qquad (4.7)$$

The total sample size is partitioned between the first and second samples by

$$n_1 = \lambda N \qquad (4.8)$$

and

$$n_2 = (1 - \lambda)N, \qquad (4.9)$$

where $0 < \lambda < 1$. With these substitutions, Equation (4.3) becomes

$$Q(N, \lambda) = \int_0^1 \Phi[\sqrt{\lambda/(1-\lambda)}\Phi^{-1}(u) + (\delta/\sigma)\sqrt{\lambda N}]du \qquad (4.10)$$

and Equation (4.6) becomes

$$Q(N, \lambda) = \int_0^1 \Phi[\sqrt{(1-\lambda)/\lambda}\Phi^{-1}(u) + (\delta/\sigma)\sqrt{(1-\lambda)N}]du. \qquad (4.11)$$

Equations (4.10) and (4.11) are transformed expressions of Equations (4.3) and (4.6) and yield the same PCS values. An equivalence between Equations (4.10) and (4.11) is also difficult to prove due to the integration of u and the complex form of the normal distribution. An equivalence becomes easier to see when specifying values for σ, δ, and N. The PCS values expressed in Equations (4.10) and (4.11) can be depicted by two-dimensional graphs, where λ is the independent variable. Figure 4.1 displays Equation (4.10) and Figure 4.2 displays Equation (4.11) for five specific cases, where σ/δ =.4 and N varies from 20 to 100. In both figures, λ varies from .01 to .99 to avoid singularities. Appendix G contains similar graphs for other σ/δ values.

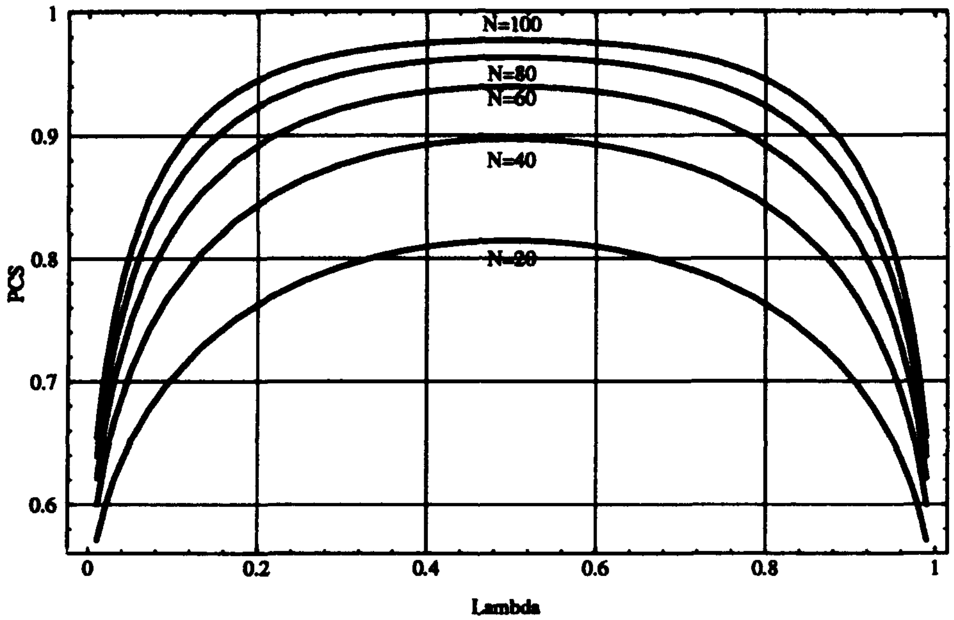

Figure 4.1 PCS vs. λ resulting from Equation (4.10) For Various N

An analysis of all graphs produced two results;

- the two functions as defined by Equations (4.10) and (4.11) appear to be equal for a fixed σ and δ (at least for the N values between 20 and 100), and
- the maximum Q values occur when λ is .5 ($n_1 = n_2$) regardless of N. This result is discussed in a later section.

This demonstrates that for the case of unequal sample sizes of k=2 and t=1, the PCS is a function of the total sample size, N, and its allocation between the two samples. Graphs for specific values help show the conjecture that any random assignment of n_1 and n_2 values to the ordered populations produces the same PCS values. Although the conjecture appears to be true for the k =2 and t =1 case, the previous development is not a proof of this conjecture. A proof could not be accomplished with this approach due to the complexity of the involved expressions. Therefore, other attempts at proving this conjecture for the k =2 and t =1 are undertaken.

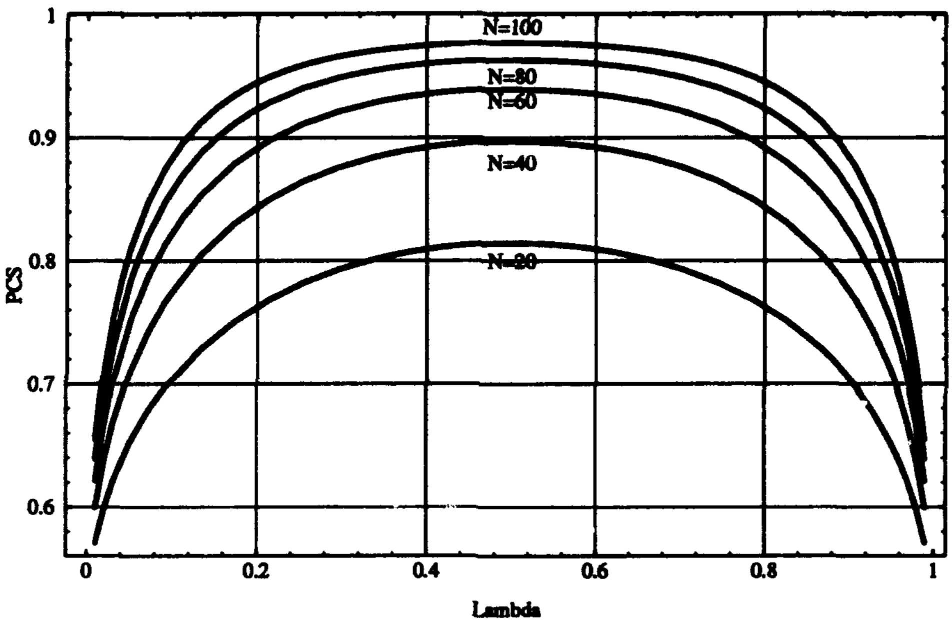

Figure 4.2 PCS vs. λ resulting from Equation (4.11) For Various N

2. Analysis Using the Uniform Distribution

The behavior of the uniform distribution for the case of k=2 and t=1 is analyzed in another attempt to analytically prove the first conjecture for a simplified case. The uniform distribution is easier to integrate and it is hoped that two results may transpire from this analysis:

- that the development of an easily integrable form for the uniform distribution might suggest a procedure to prove the results discovered for the normal distributions, and
- that the PCS function for the unequal sample size case of the uniform distribution will behave in a manner similar to the normal distribution. If the behavior is similar, then proving the first conjecture for the uniform distribution will be analytically easier and can aide in proving the conjecture for normally distributed populations.

Appendix A presents the full development and simplification of the integral expression for populations following a uniform distribution, where the sample sizes are unequal. The resulting equation from this development for the k=2 and t=1 case

$$Q = 1 - (n_{(1)}\rho^{*^{n_{(2)}}})/(n_{(1)} + n_{(2)}), \tag{4.12}$$

where $0 < \rho^* < 1$ and is specified by the experimenter. Since there are two populations, there are two sample size associations that can occur. Equations (4.1) and (4.2) describe the first possibility which can be substituted into Equation (4.12) to become

$$Q = 1 - (n_1\rho^{*^{n_2}})/(n_1 + n_2). \tag{4.13}$$

Equations (4.4) and (4.5) describe the second assignment possibility. These are substituted into Equation (4.12) to become

$$Q = 1 - (n_2\rho^{*^{n_1}})/(n_2 + n_1). \tag{4.14}$$

If we make the similar N and λ substitutions as completed in the previous development for the normal distribution, Equations (4.12) and (4.13) become

$$Q = 1 - \lambda\rho^{*^{(1-\lambda)N}} \tag{4.15}$$

and

$$Q = 1 - (1 - \lambda)\rho^{*^{\lambda N}}. \tag{4.16}$$

It is clear that Equations (4.15) and (4.16) are not equal. This is more evident when graphing these functions for a specific value of ρ^* and various values of N. Figures 4.3 and 4.4 are two-dimensional representations of PCS values as expressed by Equation (4.15) and (4.16) respectively. For each graph, N is varied from 20 to 100, λ is varied from 0 to 1, and ρ^* is fixed at .9 (arbitrarily chosen).

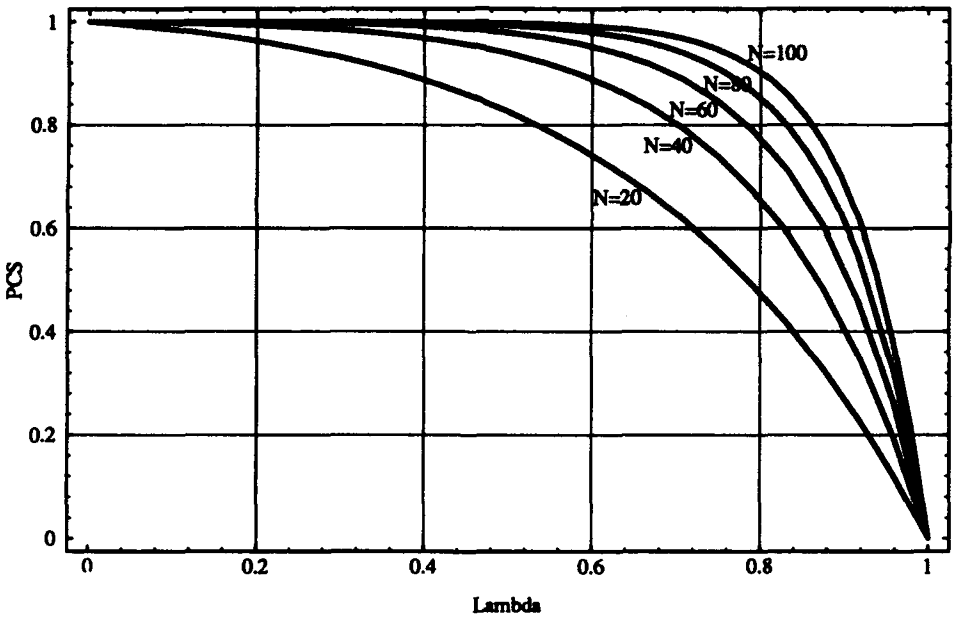

Figure 4.3 PCS vs. λ Resulting From Equation (4.15) For Various N

For unequal sample size cases of k=2 and t=1, the uniform distribution does not demonstrate the same properties as the normal distribution. Unlike the normal, the uniform distribution graphs show a dependence of the PCS on the sample size association. Clearly, using the uniform distribution is not a fruitful approach to proving the sample size association conjecture. However, it does indicate that the conjectured property of randomly assigning the sample sizes to the ordered populations without effecting the PCS might be unique to the normal distribution. Further investigation with other distributions is necessary to explore this issue.

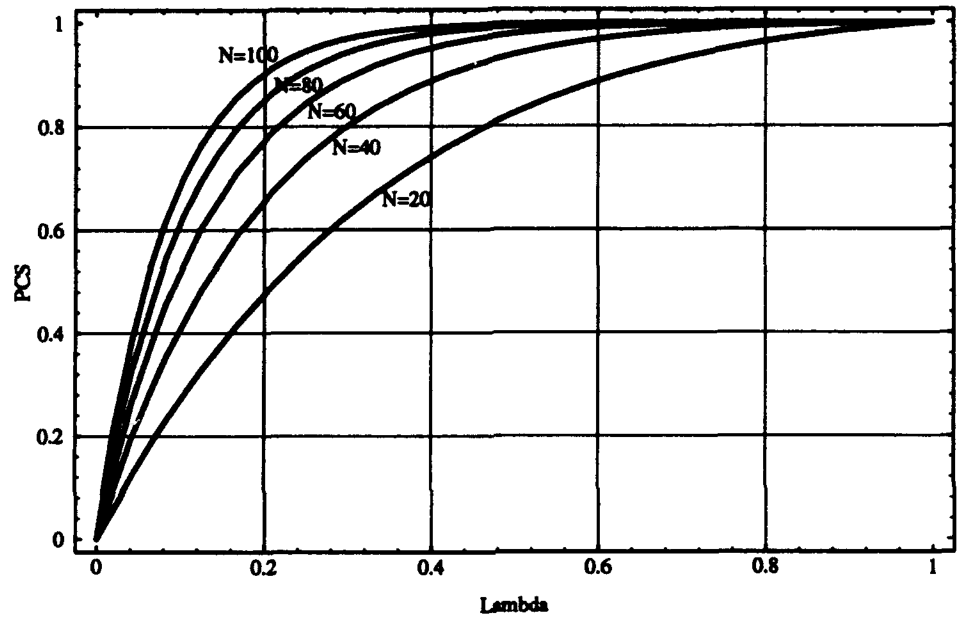

Figure 4.4 PCS vs. λ Resulting From Equation (4.16) For Various N

3. Series Expansion of Equations (4.10) and (4.11)

A third attempt was made to prove the conjecture that randomly assigning the sample sizes to the ordered populations does not effect the PCS value for the case of k=2 and t=1. This can be accomplished by proving Equations (4.10) and (4.11) are equivalent. To do this, both of these equations were expanded as a power series. If the equations are equivalent, the power series expansions should be identical. If the series expressions are not equal, the stated conjecture is clearly not true.

Mathematica was used to produce the series expansions of both Equations (4.10) and (4.11). The format for the resulting series expansions were not the same and they were much more complicated than expected. In fact, none of the terms, neither the coefficients to each factor nor the factors themselves, matched. Although inconclusive, further comparison and evaluation of the power series expansion might be worth pursuing.

4.2.4.2 Conjecture 2:. The maximum PCS occurs when the sample sizes are equal.

This second conjecture was noted previously when examining the two-dimensional graphs of Equations (4.10) and (4.11). Figures 4.1 and 4.2 reveal that the maximum PCS value for all the computed N 0 when λ=.5.

In addition, PCS values were obtained from several unequal sample size cases and compared to those obtained from cases of equal sample sizes. In every comparison where the total sample size, N, remained constant, cases involving equal samples produced the highest PCS value. One such comparision is depicted in Table 4.8. Although the table compares cases involving k =2, t =1 and N =20, the conjecture applies to other k, t, and N values as well. Appendix G contains cases which further support this conjecture.

The experimenter ideally wants a common sample size among all populations to achieve the maximum PCS. If sample sizes are not equal for a fixed N, then a lower PCS is expected. This result is not analytically proven in this thesis. However, an initial approach might involve computing the second derivatives of Equations (4.10) and (4.11) with respect to λ for the case of k =2 and t =1.

4.2.4.3 Conjecture 3:. For cases other than even k values where $t = k/2$ and a single population contains less than the nominal sample size, the PCS

Table 4.8 PCS Values as a Result of Keeping N constant and Varying the Portion Size of $n_{(1)}$ and $n_{(2)}$ for Unequal Sample Size Populations in the Case of k =2, t =1. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	*PCS*
10	10	0.814453
11	9	0.813252
12	8	0.809582
13	7	0.803234
14	6	0.793822
15	5	0.780711
16	4	0.762863
17	3	0.738507
18	2	0.704247
19	1	0.651684

value is dependent on the association of the unequal sample sizes to the ordered populations.

The first conjecture stated that for the $t = k/2$ case, an arbitrary assignment of one unequal sample size to any of the competing populations does not effect the PCS. However, this conjecture does not appear to be true for other cases of k and t. For these other cases, a sample size can be initially assigned to an ordered population and then reassigned to another ordered population, potentially producing a different PCS value.

The ideal experimental situation involves populations represented by equal sample sizes. It can be demonstrated that the PCS varies when slightly perturbing the ideal experiment through the loss of sample observations. By varying the population assignments of the smaller sample sizes and computing the PCS, the effects of the random sample size associations on the PCS value were observed. The three experimental situations that were created to make this analysis include;

- losing one observation from one population sample,
- losing one observation from each of two population samples, and

• losing two observations from one population sample.

For the specific samples sizes and δ/σ ratio, Tables 4.9 and 4.10 show that the PCS varies depending on the sample size association for the specific cases where $k=3$, $t=1$, and $k=5$, $t=2$, respectively. Appendix H, Tables H.1 through H.7 also demonstrate this conjecture for a variety of cases.

Table 4.9 PCS Values as a Result of Varying Sample Size Association for Unequal Sample Size Populations in the Case of $k=3$, $t=1$. Note: $\delta/\sigma=.4$ For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	*PCS*
14	15	15	0.773356
15	14	15	0.773356
15	15	14	0.772478
14	14	15	0.769550
14	15	14	0.768738
15	14	14	0.768738
13	15	15	0.769064
15	13	15	0.769064
15	15	13	0.767318

In analyzing the numerical values, it was discovered that every possible sample size association did not produce a unique PCS value. Sample sizes that are assigned to the first $k-t$ worst populations can be reassigned to any of these $k-t$ populations without varying the PCS. Also, sample sizes assigned to the last t best populations can be reassigned to any of these t populations and maintain the same PCS value. Therefore, the PCS will vary when a sample size reassignment involves a switch from the worst $k-t$ populations to the best t populations and vice versa.

For the tested cases, the conjecture that the PCS value is dependent on the sample size association is important in designing the QuickBASIC computer program used to calculate the PCS values. When sample sizes are not equal, an experimenter might be interested in either the range of the PCS values or the worst possible PCS. If it had been found that every possible sample size association produced different PCS values, a computer program would need to calculate the PCS for all combinations, using extensive computer resources. Since it was observed that the PCS values differ only when the sample size is reassigned from the first $k-t$ to the last t populations and vice versa, many computer calculations are eliminated, making a more efficient computer program.

Table 4.10 PCS Values as a Result of Varying Sample Size Association for Unequal Sample Size Populations in the Case of k =5, t =2. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	*PCS*
14	15	15	15	15	0.553877
15	14	15	15	15	0.553877
15	15	14	15	15	0.553877
15	15	15	14	15	0.553634
15	15	15	15	14	0.553634
14	14	15	15	15	0.550191
14	15	14	15	15	0.550191
15	14	14	15	15	0.550191
14	15	15	14	15	0.549966
15	14	15	14	15	0.549966
15	15	14	14	15	0.549966
14	15	15	15	14	0.549966
15	14	15	15	14	0.549966
15	15	14	15	15	0.549966
15	15	15	14	14	0.549713
13	15	15	15	15	0.549786
15	13	15	15	15	0.549786
15	15	13	15	15	0.549786
15	15	15	13	15	**
15	15	15	15	13	**

** Computer software could not provide a numerical answer for an unknown reason.

If an experimenter was only concerned with the worst possible PCS, a procedure which determines the exact sample size association needed to produce the minimum PCS could additionally improve the efficiency of the QuickBASIC computer program. Upon examining the data for the smallest PCS, the sample size associations to the ordered populations were observed. For cases involving k populations and t best populations, where $t<k/2$, and one unequal sample size is considered, the smallest PCS 0 when the smallest sample size was assigned to one of the t best populations. Similarly, for cases where $t>k/2$, the smallest PCS occured when the smallest sample was assigned to one of the worst populations. Appendix H contains more cases which further support this conjecture. A generalized proof is needed to solidify the argument.

4.2.4.4 Conjecture 4:. For all general cases, minor changes in one or two of the sample sizes (1 or 2 less observations) do not greatly effect the PCS.

Table 4.9, Table 4.10 and the tables located in Appendix H show the close agreement of PCS values for minor changes in one or two sample observations. When the range of possible PCS values is small for the unequal sample size case, Gibbons, Olkin, and Sobel's (GOS) procedure for approximating a common sample size should provide a reasonably accurate estimate of the PCS [4:50-51]. An option to provide approximations such as the GOS procedure via the QuickBASIC menu would enhance the developed computer program. Currently, it takes over thirty minutes to calculate all the PCS values for the case of unequal sample size, $k=6$ and $t=1$. An approximation option could simplify the unequal sample size computation to one calculation, but such an option is not yet available.

4.3 Chapter Summary

The computer program developed in this thesis can readily evaluate the integral expression associated with ranking and selection problems involving the indifference-zone formulation for normally distributed populations having common variance. For equal sample size cases, computer implementation is consistent with published tables. Other validation methods suggest this to be true for cases of unequal sample size, although a complete validation or proof was not accomplished. Conjectures were made based on empirical observations for the unequal sample size case. If eventually proven, these conjectures could prove useful in improving the efficiency of the computer program.

V. *Summary, Conclusions, and Recommendations*

5.1 Summary

This thesis focused on the indifference-zone formulation of the ranking and selecting problem for normally distributed populations with known variances. The research objective was to develop computer software program to solve the parameters of the indifference-zone integral expression presented by Barr and Rizvi for equal and unequal sample sizes. A computer program was desired for four reasons:

- The integral expression for the normal distribution is too complex to calculate by hand.
- Published tables include selected solutions to the integral expression for various ranking and selecting problems but these tables are dispersed throughout the literature; They are not available in any single convenient reference.
- Interpolation is sometimes necessary when using the published tables.
- Published tables only provide approximate solutions for unequal sample size cases.

The computer program was developed using *Mathematica* and QuickBASIC. *Mathematica* provides the intensive computational capability needed to numerically evaluate statistical distributions and integral expressions, while the QuickBASIC program creates an interactive interface between the user and *Mathematica.* While the program relies on *Mathematica* and QuickBASIC, it does not require user familiarity with either software package.

The development consisted of incorporating the integral expressions for both the equal and unequal sample size cases into the computer program. In their paper, Barr and Rizvi developed the integral expression for the case of comparing populations with a common sample size. The normal distribution was applied to this expression and simplified in order to create a *Mathematica* computational file. For the populations of unequal sample sizes, an expression first needed to be developed, then simplified for *Mathematica* implementation.

Once implemented, the *Mathematica* computations needed validation. The equal sample size computation could be easily validated using a published table. However, validating the unequal sample size computation proved difficult. Several methods were introduced to partially validate this case.

Finally, cases involving populations with unequal sample sizes were empirically analyzed. Based on the observed numerical values of the PCS, conjectures were developed which, if proven to be true, might improve the efficiency of the computer program. These conjectures are summarized below;

- Conjecture 1: When one sample size is smaller than the rest, randomly assigning the reduced sample size to the various populations produces the same PCS value for the cases of k =2, 4, or 6 and $t = k/2$.
- Conjecture 2: The maximum PCS occurs when the sample sizes are equal.
- Conjecture 3: For cases other than even k values where $t = k/2$ and a single population contains less than the nominal sample size, the PCS value is dependent on the association of the unequal sample sizes to the ordered populations.
- Conjecture 4: For all general cases, minor changes in one or two of the sample sizes does not greatly effect the PCS.

5.2 Conclusions

Based on this research effort, several conclusions were made.

1. A computer program, based on existing software, can be implemented to handle ranking and selection problems involving normally distributed populations of known variance and equal or unequal sample sizes. *Mathematica* can solve

the indifference-zone integral expression for a variety of experimental goals including:

- determining the common number of observations needed from each population to guarantee a PCS (equal sample size cases only),
- determining the PCS, given the sample sizes,
- determining the minimum indifference parameter, and
- determining a change in the population standard deviation for sensitivity analysis purposes (for equal sample size cases only).

2. The *Mathematica* computations that were created to solve the equal sample size integral expression for the above mentioned parameters were validated based on Bechhofer's published table [3:16-39].

3. The *Mathematica* computations that were created to solve the integral expression for unequal samples were only partially validated for particular cases. Confidence increased when the PCS values computed with the unequal sample size option compared well with

   - PCS values computed with the equal sample size option,
   - other computer software programs, and
   - published approximating methods.

4. Most empirical observations found for unequal sample size cases are difficult to prove for general cases. Therefore, conjectures were made. It is only hypothesized that these conjectures hold true beyond the cases that were tested in this thesis.

5. The QuickBASIC interface program could be made more efficient based on the truth of the stated conjectures for the unequal sample size cases.

6. The QuickBASIC interface program and its menu-driven display is fully capable of creating *Mathematica* input files and could be extended for other ranking and selection problems.

5.3 Recommendations

Several recommendations could either enhance the findings of this thesis or provide the basis for additional research.

1. More numerical results from unequal sample size cases should be examined to either support or disprove the stated conjectures.
2. Attention should be given to proving the general cases for the stated conjectures made in unequal sample size cases. For instance;
   - To prove Conjecture 1, distributional forms other then the normal and uniform should be examined to determine if this conjecture is unique to the normal. Also, continuation of a series expansion of Equations (4.9) and (4.10) to prove an equivalence between the two equations is worth pursuing.
   - Also, to prove Conjecture 2, the second derivative of Equations (4.10) and (4.11) should be analyzed to find the maximum PCS as a function of λ.
3. The QuickBASIC computer program could be improved. Suggested improvements are listed;
   - Include a display which shows the numerical input values for the unequal sample size selections.
   - Include a recover option for the unequal sample size selections.
   - Make the menu-driven program more aesthetic to the user.
   - Incorporate other menu options into the computer program (*e.g.* Normal-Ranking Variances).

- Include an option for the user to choose an answer which is exact or approximate. For unequal sample size cases, an approximate answer may take less computational time and provide a solution suitable to the user. Bechhofer's table could be incorporated into the computer program, minimizing calculation.
- Based on Conjecture 3, incorporate a procedure which directly determines the minimum PCS for unequal sample size populations.

4. A QuickBASIC program should be created which can read the *Mathematica* results file and format it into a table.
5. Other distributional forms, such as the uniform and exponential, should be incorporated into the computer program to provide solutions to a wider scope of ranking and selection problems.

Appendix A. Indifference-Zone Formulation Derivation Involving Uniformly Distributed Populations for Equal and Unequal Sample Sizes

A.1 Equal Sample Sizes

To derive an indifference-zone formulation for uniform populations of equal sample sizes, we can directly apply either of the general formulas developed in Chapter 2 (Equations (2.16) or (2.21)). Equation (2.21) is applied and is repeated here for convenience:

$$Q = t \int_{-\infty}^{\infty} F^{k-t}(y;\theta)[1 - F(y;\psi(\theta))]^{t-1} dF(y;\psi(\theta)). \tag{A.1}$$

Consider a comparison of k uniformly distributed populations in which the goal is to select t of them. A common sample size is taken from each of the populations to guarantee a correct selection with a probability of at least P^*. P^* provides the lower bound for Q (Q has to be at least P^* or better). Since the uniform distribution belongs to a scale parameter family, the CDF can be expressed as

$$F(y;\theta) = F(y/\theta;1). \tag{A.2}$$

With this substitution, Equation (A.1) becomes

$$Q = t \int_{-\infty}^{\infty} F^{k-t}(y/\theta;1)[1 - F(y/\psi(\theta);1)]^{t-1} dF(y/\psi(\theta);1). \tag{A.3}$$

Because the uniform distribution is a member of a scale parameter family, we can choose $\psi(\theta)=\theta/\rho^*$, where ρ^* is specified by the experimenter ($0 < \rho^* < 1$, since $\theta < \psi(\theta)$). Making this substitution, Q becomes a function of θ. Therefore, an expression for any distribution with equal sample sizes that is a member of a scale parameter family can be expressed as

$$Q = t \int_{-\infty}^{\infty} F^{k-t}(y/\theta;1)[1 - F(y\rho^*/\theta;1)]^{t-1} dF(y\rho^*/\theta;1). \tag{A.4}$$

If we let $X_1, \ldots, X_n$ be the sample of n independent observations from a uniform distribution, the PDF is

$$g(x/\theta; 1) = \begin{cases} 1/\theta & 0 < x < \theta, 0 < \theta \\ 0 & \text{elsewhere} \end{cases} \tag{A.5}$$

If we consider $Y = max(X_1, \ldots, X_n)$, then the CDF for Y is

$$F(y/\theta; 1) = \begin{cases} 0 & y < 0 \\ (y/\theta)^n & 0 \leq y < \theta \\ 1 & \theta \leq y \end{cases} \tag{A.6}$$

and therefore, its PDF is

$$f(y/\theta; 1) = \begin{cases} ny^{n-1}/\theta^n & 0 < y < \theta \\ 0 & \text{elsewhere} \end{cases} \tag{A.7}$$

These distributions for the Y statistic are substituted into Equation (A.4), forcing a change in integration limits and becoming

$$Q = t \int_0^\theta (y/\theta)^{n(k-t)} [1 - (y\rho^*/\theta)^n]^{t-1} ny^{n-1} \rho^{*^n} / \theta^n dy. \tag{A.8}$$

To simplify Equation (A.8), let

$$u = (y\rho^*/\theta)^n \tag{A.9}$$

so that

$$u/\rho^{*^n} = (y/\theta)^n \tag{A.10}$$

and

$$du = ny^{n-1} \rho^{*^n} / \theta^n dy. \tag{A.11}$$

The final expression of the PCS for populations involving uniform distributions with equal sample sizes becomes

$$Q = t \int_0^1 (u/\rho^{*n})^{k-t}(1-u)^{t-1} du, \tag{A.12}$$

where Q is a strictly increasing function of n. Therefore, there is a minimum n value for which $Q \geq P^*$, the specified PCS [2:16-18]. Equation (A.12) provides an easy, integrable equation to solve for either n, the PCS, or ρ^* when all other variables are known.

A.2 Unequal Sample Sizes

We can use either Equation (2.10) or (2.18) from Chapter 2 to develop a formula for problems involving uniformly distributed populations of unequal sample sizes. Equation (2.10) is chosen and is repeated here for convenience;

$$P = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b \neq i}}^{k-t} F_{Y_{(b)}}(y) \prod_{a=k-t+1}^{k} [1 - F_{Y_{(a)}}(y)] f_{Y_{(i)}}(y) dy. \tag{A.13}$$

The CDF and the PDF expressions, which are eventually substituted into the above equation are listed in Equations (A.14) through (A.17);

$$F_{Y_{(b)}}(y) = \begin{cases} 0 & y < 0 \\ (y/\theta_{[b]})^{n_{(b)}} & 0 \leq y < \theta_{[b]} \\ 1 & \theta_{[b]} \leq y \end{cases} \tag{A.14}$$

$$F_{Y_{(a)}}(y) = \begin{cases} 0 & y < 0 \\ (y/\theta_{[a]})^{n_{(a)}} & 0 \leq y < \theta_{[a]} \\ 1 & \theta_{[a]} \leq y \end{cases} \tag{A.15}$$

$$F_{Y_{(i)}}(y) = \begin{cases} 0 & y < 0 \\ (y/\theta_{[i]})^{n_{(i)}} & 0 \le y < \theta_{[i]} \\ 1 & \theta_{[i]} \le y \end{cases} \tag{A.16}$$

$$f_{Y_{(i)}}(y) = \begin{cases} n_{(i)} y^{n_{(i)}-1}/\theta_{[i]}^{n_{(i)}} & 0 < y < \theta_{[i]} \\ 0 & \text{elsewhere} \end{cases} \tag{A.17}$$

Before substitution, we can take the limit of the probability in Equation (A.13) as we let the parameters of the first $k-t$ populations approach θ, and the parameters of the last t populations approach a function of θ, or θ/ρ^*. This limit can be expressed as

$$Q = \lim_{\substack{\theta_{[b]}\to\theta \\ \theta_{[a]}\to\theta/\rho^*}} \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b\neq i}}^{k-t} F_{Y_{(b)}}(y) \prod_{a=k-t+1}^{k} [1 - F_{Y_{(a)}}(y)] f_{Y_{(i)}}(y)dy. \tag{A.18}$$

After taking the limit, the CDF's and PDF's can be expressed as functions of specific parameters, namely θ, n and θ/ρ^*;

$$Q = \sum_{i=1}^{k-t} \int_{-\infty}^{\infty} \prod_{\substack{b=1 \\ b\neq i}}^{k-t} F(y;\theta,n_{(b)}) \prod_{a=k-t+1}^{k} [1 - F(y;\theta/\rho^*,n_{(a)})] f(y;\theta,n_{(i)})dy. \tag{A.19}$$

When making the substitutions expressed in Equations (A.14) through (A.17), the limits of integration change and Equation (A.19) becomes

$$Q = \sum_{i=1}^{k-t} \int_{0}^{\theta} \prod_{\substack{b=1 \\ b\neq i}}^{k-t} (y/\theta)^{n_{(b)}} \prod_{a=k-t+1}^{k} [1 - (y\rho^*/\theta)^{n_{(a)}}] n_{(i)} y^{n_{(i)}-1}/\theta^{n_{(i)}} dy. \tag{A.20}$$

A dummy variable u can be defined as

$$u = y/\theta, \tag{A.21}$$

where

$$y = u\theta \tag{A.22}$$

and

$$du = (1/\theta)dy. \tag{A.23}$$

These substitutions change the limits of integration and Equation (A.20) becomes

$$Q = \sum_{i=1}^{k-t} \int_0^1 \prod_{\substack{b=1 \\ b \neq i}}^{k-t} (u)^{n_{(b)}} \prod_{a=k-t+1}^{k} [1 - (u\rho^*)^{n_{(a)}}](n_{(i)}(u\theta)^{n_{(i)}-1}/\theta^{n_{(i)}-1})du. \tag{A.24}$$

Terms cancel and the equation can be simplified to the general formula of

$$Q = \sum_{i=1}^{k-t} \int_0^1 \prod_{\substack{b=1 \\ b \neq i}}^{k-t} (u)^{n_{(b)}} \prod_{a=k-t+1}^{k} [1 - (u\rho^*)^{n_{(a)}}] n_{(i)}(u)^{n_{(i)}-1} du. \tag{A.25}$$

A.2.1 Case when t=1. For the particular case of any k and t=1, Equation (A.25) can be expressed as

$$Q = \sum_{i=1}^{k-1} \int_0^1 \prod_{\substack{b=1 \\ b \neq i}}^{k-1} (u)^{n_{(b)}} \prod_{a=k}^{k} [1 - (u\rho^*)^{n_{(a)}}] n_{(i)}(u)^{n_{(i)}-1} du. \tag{A.26}$$

This simplifies to

$$Q = \sum_{i=1}^{k-1} \int_0^1 [1 - (u\rho^*)^{n_{(k)}}] n_{(i)}(u)^{n_{(i)}-1} \prod_{\substack{b=1 \\ b \neq i}}^{k-1} (u)^{n_{(b)}} du \tag{A.27}$$

which is the simplest form that this expression takes for this particular case.

A.2.2 Case when k=2 and t=1. For this case, Equation (A.26) can be represented as

$$Q = \sum_{i=1}^{1} \int_0^1 \prod_{\substack{b=1 \\ b \neq i}}^{1} (u)^{n_{(b)}} \prod_{a=2}^{2} [1 - (u\rho^*)^{n_{(a)}}] n_{(i)}(u)^{n_{(i)}-1} du. \tag{A.28}$$

This equation simplifies in the following steps;

$$Q = \int_0^1 [1 - (u\rho^*)^{n_{(2)}}] n_{(1)}(u)^{n_{(1)}-1} du, \tag{A.29}$$

$$Q = n_{(1)} \int_0^1 u^{n_{(1)}-1} [1 - (u\rho^*)^{n_{(2)}}] du, \tag{A.30}$$

$$Q = n_{(1)} \int_0^1 [u^{n_{(1)}-1} - u^{n_{(1)}-1} (u\rho^*)^{n_{(2)}}] du, \tag{A.31}$$

and

$$Q = n_{(1)} [\int_0^1 u^{n_{(1)}-1} du - {\rho^*}^{n_{(2)}} \int_0^1 u^{n_{(1)}+n_{(2)}-1} du]. \tag{A.32}$$

Integrating leads to

$$Q = n_{(1)} [(1/n_{(1)}) - {\rho^*}^{n_{(2)}} (1/(n_{(1)} + n_{(2)}))], \tag{A.33}$$

or

$$Q = 1 - (n_{(1)} {\rho^*}^{n_{(2)}})/(n_{(1)} + n_{(2)}), \tag{A.34}$$

which is the same expression as Equation (4.12) in Chapter 4.

Appendix B. Directions for Operating the Ranking and Selection Computer Software Program Using the AFIT UNIX Operating System

1. Load the QuickBASIC executable file onto a PC MS-DOS compatible computer. The file is currently titled 'Normal.exe'.
2. In the directory in which 'Normal.exe' is loaded, enter '**Normal**' at the prompt.
3. The QuickBASIC menu selection will begin. Make the menu selections and enter the numerical values as directed. The program will ask for a file name and a disk drive on which to save the created file for input into *Mathematica*;
   - If saving the input file to a floppy diskette, select the appropriate floppy drive, *d*, and go to Step 4.
   - If saving to the computer hard drive, ensure that the computer can connect to the UNIX operating system. Select the appropriate hard drive which allows this connection and go to Step 6.

<u>Loading the Input File From A Floppy Disk.</u>

4. To load the *Mathematica* input file on the SUN SPARC work stations, inject the floppy diskette into the work station and enter the following UNIX commands:
   - '**mcopy** ***d*****:ifilename ifilename**', where 'ifilename' is the name that was given for the file, and
   - '**msdos2unix ifilename ifilename**' which transfers the file text from ASCII to UNIX format.
5. Go to step 11

Transferring the Input File To A Workstation.

6. To transfer the *Mathematica* input file from the computer hard drive to the workstation, you must be in the same directory and drive as the saved input file.
7. Enter '**ftp (SUN Sparc machine name)**' for transferring files to the UNIX system. Some of the machine names used at AFIT are 'eel', 'cod', and 'alligator'.
8. The system will prompt for a username and password which should be entered by the user.
9. Enter the command '**send ifilename**' to transfer the file, where 'ifilname' is the name of the saved file.
10. Now you can either login to the UNIX operating system via the SUN Sparc workstations, or remote login from the PC that you used to transfer the input file. Either login should give a UNIX prompt.

11. The UNIX command '**math<ifilename >ofilename**' can now be entered. *Mathematica* responds to the statements in the input file and stores results in the output file named 'ofilename'. The output file will be created in the same UNIX directory that the UNIX command '**math**' is entered.
12. A UNIX prompt signifies the completion of the calculation and the output file 'ofilename' is ready to view in the UNIX operating system.

Appendix C. QuickBASIC Code for Ranking and Selection Menu-Driven Computer Program

```
REM Version 1.0 December 2 1993
DIM SS(500)
COLOR 14, 1
1 CLS
REM
REM Level I Menu
REM
LOCATE 6, 15: PRINT "RANKING AND SELECTION PROBLEMS"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Select option.  You do not have to press
 ENTER)"
LOCATE 9, 15: PRINT ""
LOCATE 10, 15: PRINT "Options for the Distributions are:"
LOCATE 11, 15: PRINT ""
LOCATE 12, 15: PRINT "  (1) Normal - Ranking Means"
LOCATE 13, 15: PRINT "* (2) Normal - Ranking Variances"
LOCATE 14, 15: PRINT "  (3) Quit"
LOCATE 15, 15: PRINT ""
LOCATE 16, 15: PRINT "(*) indicates option is not available"
LOCATE 17, 15: PRINT ""
10 LOCATE 18, 15: PRINT "Select Option:"
100 D$ = INKEY$: IF D$ = "" THEN GOTO 100
REM
REM The following questions the user whether they have made the
REM correct choice
REM
200 IF D$ = "1" THEN
        LOCATE 19, 15: PRINT "You have selected the Normal,
 Ranking Means"
        LOCATE 20, 15: PRINT "Menu.  Is this where you want to go
(Y OR N)?"
220         D$ = INKEY$: IF D$ = "" THEN GOTO 220
            IF D$ = "Y" THEN
              GOTO 300
            ELSEIF D$ = "N" THEN
              GOTO 1
            ELSE
```

```
            GOTO 220
          END IF
      ELSEIF D$ = "2" THEN
        LOCATE 19, 15: PRINT "Option not available at this time."
        LOCATE 20, 15: PRINT "Reselect Option:"
        GOTO 10
REM
REM The following QUIT option is available on most of the menu
REM choices
REM
      ELSEIF D$ = "3" THEN
        LOCATE 21, 15: PRINT "You have selected to Quit"
        LOCATE 22, 15: PRINT "Is this what you want to do
(Y or N?):"
240     D$ = INKEY$: IF D$ = "" THEN GOTO 240
        IF D$ = "Y" THEN
          STOP
        ELSEIF D$ = "N" THEN
        GOTO 1
        ELSE
        GOTO 240
        END IF
   END IF
300 CLS
REM
REM Level II Menu
REM
LOCATE 6, 15: PRINT "Normal - Ranking Means Menu"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Select option.  You do not have to press
 ENTER)"
LOCATE 9, 15: PRINT ""
LOCATE 10, 15: PRINT "Options are:"
LOCATE 11, 15: PRINT ""
LOCATE 12, 15: PRINT "  (1) Equal Sample Sizes, Known Variance"
LOCATE 13, 15: PRINT "  (2) Unequal Sample Sizes, Known Variance"
LOCATE 14, 15: PRINT "* (3) Unequal Sample Sizes, Unknown Variance"
LOCATE 15, 15: PRINT "  (4) Quit"
LOCATE 16, 15: PRINT ""
LOCATE 17, 15: PRINT " (*) indicates option is not available"
LOCATE 18, 15: PRINT ""
399 LOCATE 19, 15: PRINT "Select Option:"
```

```
400 D$ = INKEY$: IF D$ = "" THEN GOTO 400
500 IF D$ = "1" THEN
       LOCATE 20, 15: PRINT "You have selected the Equal
Sample Sizes"
       LOCATE 21, 15: PRINT "option.  Is this where you want to go
 (Y or N)?"
520        D$ = INKEY$: IF D$ = "" THEN GOTO 520
           IF D$ = "Y" THEN
             GOTO 600
           ELSEIF D$ = "N" THEN
             GOTO 300
           ELSE
             GOTO 520
           END IF
       ELSEIF D$ = "2" THEN
         LOCATE 20, 15: PRINT "You have selected the Unequal Sample
Size"
         LOCATE 21, 15: PRINT "option. Is this where you want to go
 (Y or N)?"
540        D$ = INKEY$: IF D$ = "" THEN GOTO 540
           IF D$ = "Y" THEN
             GOTO 1300
           ELSEIF D$ = "N" THEN
             GOTO 300
           ELSE
             GOTO 540
           END IF
       ELSEIF D$ = "3" THEN
         LOCATE 20, 15: PRINT "Option not available at this time."
         LOCATE 21, 15: PRINT "Reselect Option:"
         GOTO 399
       ELSEIF D$ = "4" THEN
         LOCATE 20, 15: PRINT "You have selected to Quit"
         LOCATE 21, 15: PRINT "Is this what you want to do (Y or N)?"
560        D$ = INKEY$: IF D$ = "" THEN GOTO 560
           IF D$ = "Y" THEN
             STOP
           ELSEIF D$ = "N" THEN
             GOTO 300
           ELSE
             GOTO 560
           END IF
```

```
   END IF
600 CLS
REM
REM Level III Menu
REM
LOCATE 6, 15: PRINT "Normal-Ranking Means-Equal Sample Size,
Known Variance Menu"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Select option.  You do not have to press
 ENTER)"
LOCATE 9, 15: PRINT ""
LOCATE 10, 15: PRINT "Options are:"
LOCATE 11, 15: PRINT ""
LOCATE 12, 15: PRINT "(1) Solve for Common Sample Size, n"
LOCATE 13, 15: PRINT "(2) Solve for Probability of Correct Selection,
PCS"
LOCATE 14, 15: PRINT "(3) Solve for the Indifference Parameter,
 Delta"
LOCATE 15, 15: PRINT "(4) Solve for Standard Deviation, Sigma"
LOCATE 16, 15: PRINT "(5) Quit"
LOCATE 17, 15: PRINT ""
610 LOCATE 18, 15: PRINT "Select Option:"
620 D$ = INKEY$: IF D$ = "" THEN GOTO 620
640 IF D$ = "1" THEN
     LOCATE 19, 15: PRINT "You have selected to solve for Sample
 Size."
     LOCATE 20, 15: PRINT "Is this what you want to do (Y or N)?"
642         D$ = INKEY$: IF D$ = "" THEN GOTO 642
            IF D$ = "Y" THEN
              GOTO 900
            ELSEIF D$ = "N" THEN
              GOTO 600
            ELSE
              GOTO 642
            END IF
    ELSEIF D$ = "2" THEN
        LOCATE 19, 15: PRINT "You have selected to solve for the
 PCS."
        LOCATE 20, 15: PRINT "Is this what you want to do
(Y or N)?"
644         D$ = INKEY$: IF D$ = "" THEN GOTO 644
            IF D$ = "Y" THEN
```

```
              GOTO 1000
            ELSEIF D$ = "N" THEN
              GOTO 600
            ELSE
              GOTO 644
            END IF
    ELSEIF D$ = "3" THEN
        LOCATE 19, 15: PRINT "You have selected to solve for the
indifference"
        LOCATE 20, 15: PRINT "parameter.  Is this what you want to do
(Y or N)?"
646         D$ = INKEY$: IF D$ = "" THEN GOTO 646
            IF D$ = "Y" THEN
              GOTO 1100
            ELSEIF D$ = "N" THEN
              GOTO 600
            ELSE
              GOTO 646
            END IF
    ELSEIF D$ = "4" THEN
        LOCATE 19, 15: PRINT "You have selected to solve for the
 standard"
        LOCATE 20, 15: PRINT "deviation.  Is this what you want to do
(Y or N)?"
648         D$ = INKEY$: IF D$ = "" THEN GOTO 648
            IF D$ = "Y" THEN
              GOTO 1200
            ELSEIF D$ = "N" THEN
              GOTO 600
            ELSE
              GOTO 648
            END IF
    ELSEIF D$ = "5" THEN
        LOCATE 19, 15: PRINT "You have selected to Quit"
        LOCATE 20, 15: PRINT "Is this what you want to do (Y or N)?"
650         D$ = INKEY$: IF D$ = "" THEN GOTO 650
            IF D$ = "Y" THEN
              STOP
            ELSEIF D$ = "N" THEN
              GOTO 600
            ELSE
              GOTO 650
```

```
            END IF
        END IF
900 CLS
REM
REM Level IV Menu solving for n.  It asks for numerical answers
REM from the user.
REM
LOCATE 6, 15: PRINT "Normal-Ranking Means-Equal Sample Sizes,
 Known Variance - Solve for n"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Input value and hit the ENTER key)"
LOCATE 9, 15: PRINT ""
INPUT "(1) Enter indifference parameter, delta"; del
INPUT "(2) Enter standard deviation, sigma"; sig
INPUT "(3) Enter number of populations to be ranked"; k
INPUT "(4) Enter number of best populations desired"; t
INPUT "(5) Enter desired probability of correct selection"; PCS
REM
REM The user has the option of either entering their own
REM search values for n or having Mathematica do this
REM for them.  If letting the software compute values,
REM then another Mathematica computation file is called
REM upon to compute two approximate search values.
REM This option is available for equal sample sizes
REM when solving for delta and sigma also.
REM
LOCATE 15, 10: PRINT "Do you want to enter your own
 search values
for n (Y or N)?"
LOCATE 16, 10: PRINT "(If 'N' then the program computes values
 for the
 search)"
904  DS$ = INKEY$: IF DS$ = "" THEN GOTO 904
     IF DS$ = "Y" THEN
     INPUT "(6) Enter first search value for n"; fst
     INPUT "(7) Enter second search value for n"; sec
     ELSEIF DS$ = "N" THEN
     GOTO 905
     ELSE
     GOTO 904
     END IF
REM
```

```
REM The user has the option to save to a specific drive and
REM file name.
REM
905  LOCATE 19, 15: PRINT "(8) Enter the drive you want the
exported file on,"
     LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
910 INPUT DN$
   IF DN$ = "" THEN
      DN$ = "C"
   END IF
LOCATE 21, 15: PRINT "(9) Enter the name of the data file,"
LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
LOCATE 23, 15: PRINT "    (default is norm1):"
920 INPUT NF$
    IF NF$ = "" THEN
       NF$ = "norm1"
    END IF
930 IN$ = DN$ + ":" + NF$ + ".txt"
940 CLS
REM
REM  The following screen allows the user to check the numerical
REM values that he entered and change them if necessary.
REM
941 LOCATE 6, 15: PRINT "The following are the values that you
 assigned:"
LOCATE 7, 15: PRINT "(1) delta="; del
LOCATE 8, 15: PRINT "(2) sigma="; sig
LOCATE 9, 15: PRINT "(3) number of populations="; k
LOCATE 10, 15: PRINT "(4) number of best populations="; t
LOCATE 11, 15: PRINT "(5) desired PCS="; PCS
   IF DS$ = "Y" THEN
      LOCATE 12, 15: PRINT "(6) first search value for n=";
 fst
      LOCATE 13, 15: PRINT "(7) second search value for n=";
 sec
   ELSE
      GOTO 942
   END IF
942 LOCATE 14, 15: PRINT "(8) drive to store file="; DN$
    LOCATE 15, 15: PRINT "(9) name of exported file="; NF$
LOCATE 16, 15: PRINT ""
LOCATE 17, 15: PRINT "Do you want to make any changes (Y or N
```

```
or Q to Quit)?"
950 D$ = INKEY$: IF D$ = "" THEN GOTO 950
IF D$ = "Y" THEN
       LOCATE 18, 15: PRINT "Which value do you want to change?"
       LOCATE 19, 15: PRINT "(enter number in parenthesis)"
952    DV$ = INKEY$: IF DV$ = "" THEN GOTO 952
       IF DV$ = "1" THEN
         CLS
         INPUT "(1) Enter indifference parameter, delta"; del
         GOTO 940
       ELSEIF DV$ = "2" THEN
         CLS
         INPUT "(2) Enter standard deviation, sigma"; sig
         GOTO 940
       ELSEIF DV$ = "3" THEN
         CLS
         INPUT "(3) Enter number of populations to be ranked"; k
         GOTO 940
       ELSEIF DV$ = "4" THEN
         CLS
         INPUT "(4) Enter number of best populations desired"; t
         GOTO 940
       ELSEIF DV$ = "5" THEN
         CLS
         INPUT "(5) Enter desired probability of correct selection";
 PCS
         GOTO 940
       ELSEIF DV$ = "6" THEN
         CLS
         INPUT "(6) Enter first search value for n"; fst
         GOTO 940
       ELSEIF DV$ = "7" THEN
         CLS
         INPUT "(7) Enter second search value for n"; sec
         GOTO 940
       ELSEIF DV$ = "8" THEN
         CLS
         LOCATE 19, 15: PRINT "(8) Enter the drive you want the
exported file on,"
         LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
954      DN$ = INKEY$: IF DN$ = "" THEN 954
956      IN$ = DN$ + ":" + NF$ + ".txt"
```

```
          GOTO 940
        ELSEIF DV$ = "9" THEN
          CLS
          LOCATE 21, 15: PRINT "(9) Enter the name of the data file,"
          LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
          LOCATE 23, 15: PRINT "    (default is Norm1):"
958       INPUT NF$
960       IN$ = DN$ + ":" + NF$ + ".txt"
          GOTO 940
        ELSE
          GOTO 950
        END IF
    ELSEIF D$ = "N" THEN
       GOTO 970
    ELSEIF D$ = "Q" THEN
       GOTO 5000
    ELSE
       GOTO 952
     END IF
970 OPEN IN$ FOR OUTPUT AS #1
REM
REM The following is the file created to input into the
REM Mathematica program.
REM
PRINT #1, "<<normeq"
PRINT #1, "delta =", del
PRINT #1, "sigma =", sig
PRINT #1, "k=", k
PRINT #1, "t=", t
PRINT #1, "pr=", PCS
REM
REM If the user wants Mathematica to compute search values then
REM the following file is invoked.
REM
  IF DS$ = "N" THEN
     PRINT #1, "<<nest"
  ELSE
REM
REM Otherwise, the user inputs the search values for n.
REM
     PRINT #1, "fst[t,k]=", fst
     PRINT #1, "sec[t,k]=", sec
```

```
  END IF
PRINT #1, "R[t,k,pr]//N"
CLOSE #1
GOTO 5000
1000 CLS
REM
REM Level IV Menu, solving for PCS.  The rest of the equal sample
REM size menus and options are similar to the previous,
REM solve for n menu and options.
REM
LOCATE 6, 15: PRINT "Normal-Ranking Means-Equal Sample Sizes,
Known Variance-Solve for PCS"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Input value and hit the ENTER key)"
LOCATE 9, 15: PRINT ""
INPUT "(1) Enter indifference parameter, delta"; del
INPUT "(2) Enter standard deviation, sigma"; sig
INPUT "(3) Enter number of populations to be ranked"; k
INPUT "(4) Enter number of best populations desired"; t
INPUT "(5) Enter sample size"; n
LOCATE 19, 15: PRINT "(6) Enter the drive you want the exported
 file on,"
LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
1010 INPUT DN$
   IF DN$ = "" THEN
      DN$ = "C"
   END IF
LOCATE 21, 15: PRINT "(7) Enter the name of the data file,"
LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
LOCATE 23, 15: PRINT "    (default is norm1):"
1020 INPUT NF$
    IF NF$ = "" THEN
       NF$ = "norm1"
    END IF
1030 IN$ = DN$ + ":" + NF$ + ".txt"
1040 CLS
1041 LOCATE 6, 15: PRINT "The following are the values that you
 assigned:"
LOCATE 7, 15: PRINT "(1) delta="; del
LOCATE 8, 15: PRINT "(2) sigma="; sig
LOCATE 9, 15: PRINT "(3) number of populations="; k
LOCATE 10, 15: PRINT "(4) number of best populations="; t
```

```
LOCATE 11, 15: PRINT "(5) sample size="; n
LOCATE 12, 15: PRINT "(6) drive to store file="; DN$
LOCATE 13, 15: PRINT "(7) name of exported file="; NF$
LOCATE 14, 15: PRINT ""
LOCATE 15, 15: PRINT "Do you want to make any changes (Y or N
or Q to Quit)?"
1050 D$ = INKEY$: IF D$ = "" THEN GOTO 1050
IF D$ = "Y" THEN
       LOCATE 16, 15: PRINT "Which value do you want to change?"
       LOCATE 17, 15: PRINT "(enter number in parenthesis)"
1052    DV$ = INKEY$: IF DV$ = "" THEN GOTO 1052
       IF DV$ = "1" THEN
         CLS
         INPUT "(1) Enter indifference parameter, delta"; del
         GOTO 1040
       ELSEIF DV$ = "2" THEN
         CLS
         INPUT "(2) Enter standard deviation, sigma"; sig
         GOTO 1040
       ELSEIF DV$ = "3" THEN
         CLS
         INPUT "(3) Enter number of populations to be ranked"; k
         GOTO 1040
       ELSEIF DV$ = "4" THEN
         CLS
         INPUT "(4) Enter number of best populations desired"; t
         GOTO 1040
       ELSEIF DV$ = "5" THEN
         CLS
         INPUT "(5) Enter common sample size"; n
         GOTO 1040
       ELSEIF DV$ = "6" THEN
         CLS
         LOCATE 19, 15: PRINT "(6) Enter the drive you want the
 exported file on,"
         LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
1054      DN$ = INKEY$: IF DN$ = "" THEN 1054
1056      IN$ = DN$ + ":" + NF$ + ".txt"
         GOTO 1040
       ELSEIF DV$ = "7" THEN
         CLS
         LOCATE 21, 15: PRINT "(7) Enter the name of the data file,"
```

```
        LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
        LOCATE 23, 15: PRINT "    (default is Norm1):"
1058     INPUT NF$
1060     IN$ = DN$ + ":" + NF$ + ".txt"
        GOTO 1040
      ELSE
        GOTO 1050
      END IF
   ELSEIF D$ = "N" THEN
      GOTO 1070
   ELSEIF D$ = "Q" THEN
      GOTO 5000
   ELSE
      GOTO 1052
    END IF
1070 OPEN IN$ FOR OUTPUT AS #2
PRINT #2, "<<normeq"
PRINT #2, "delta=", del
PRINT #2, "sigma=", sig
PRINT #2, "k=", k
PRINT #2, "t=", t
PRINT #2, "n=", n
PRINT #2, "Q[n,t,k]//N"
CLOSE #2
GOTO 5000
1100 CLS
LOCATE 6, 15: PRINT "Normal-Ranking Means-Equal Sample Sizes,
Known Variance-Solve for Delta"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Input value and hit the ENTER key)"
LOCATE 9, 15: PRINT ""
INPUT "(1) Enter standard deviation, sigma"; sig
INPUT "(2) Enter sample size"; n
INPUT "(3) Enter number of populations to be ranked"; k
INPUT "(4) Enter number of best populations desired"; t
INPUT "(5) Enter desired probability of correct selection"; PCS
LOCATE 15, 10: PRINT "Do you want to enter your own search values
 for delta (Y or N)?"
LOCATE 16, 10: PRINT "(If 'N' then the program computes values for
the search)"
1104  DS$ = INKEY$: IF DS$ = "" THEN GOTO 1104
     IF DS$ = "Y" THEN
```

```
     INPUT "(6) Enter first search value for delta"; fst
     INPUT "(7) Enter second search value for delta"; sec
     ELSEIF DS$ = "N" THEN
     GOTO 1105
     ELSE
     GOTO 1104
     END IF
1105 LOCATE 19, 15: PRINT "(8) Enter the drive you want the
 exported file on,"
LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
1110 INPUT DN$
   IF DN$ = "" THEN
      DN$ = "C"
   END IF
LOCATE 21, 15: PRINT "(9) Enter the name of the data file,"
LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
LOCATE 23, 15: PRINT "    (default is norm1):"
1120 INPUT NF$
    IF NF$ = "" THEN
       NF$ = "norm1"
    END IF
1130 IN$ = DN$ + ":" + NF$ + ".txt"
1140 CLS
1141 LOCATE 6, 15: PRINT "The following are the values that you
assigned:"
LOCATE 7, 15: PRINT "(1) sigma="; sig
LOCATE 8, 15: PRINT "(2) sample size="; n
LOCATE 9, 15: PRINT "(3) number of populations="; k
LOCATE 10, 15: PRINT "(4) number of best populations="; t
LOCATE 11, 15: PRINT "(5) desired PCS="; PCS
   IF DS$ = "Y" THEN
      LOCATE 12, 15: PRINT "(6) first search value for delta=";
 fst
      LOCATE 13, 15: PRINT "(7) second search value for delta=";
 sec
   ELSE
      GOTO 1142
   END IF
1142 LOCATE 14, 15: PRINT "(8) drive to store file="; DN$
LOCATE 15, 15: PRINT "(9) name of exported file="; NF$
LOCATE 16, 15: PRINT ""
LOCATE 17, 15: PRINT "Do you want to make any changes (Y or N
```

```
or Q to Quit)?"
1150 D$ = INKEY$: IF D$ = "" THEN GOTO 1150
IF D$ = "Y" THEN
       LOCATE 18, 15: PRINT "Which value do you want to change?"
       LOCATE 19, 15: PRINT "(enter number in parenthesis)"
1152    DV$ = INKEY$: IF DV$ = "" THEN GOTO 1152
       IF DV$ = "1" THEN
         CLS
         INPUT "(1) Enter standard deviation, sigma"; sig
         GOTO 1140
       ELSEIF DV$ = "2" THEN
         CLS
         INPUT "(2) Enter sample size, n"; n
         GOTO 1140
       ELSEIF DV$ = "3" THEN
         CLS
         INPUT "(3) Enter number of populations to be ranked"; k
         GOTO 1140
       ELSEIF DV$ = "4" THEN
         CLS
         INPUT "(4) Enter number of best populations desired"; t
         GOTO 1140
       ELSEIF DV$ = "5" THEN
         CLS
         INPUT "(5) Enter desired probability of correct selection";
 PCS
         GOTO 1140
       ELSEIF DV$ = "6" THEN
         CLS
         INPUT "(6) Enter first search value for delta"; fst
         GOTO 1140
       ELSEIF DV$ = "7" THEN
         CLS
         INPUT "(7) Enter second search value for delta"; sec
         GOTO 1140
       ELSEIF DV$ = "8" THEN
         CLS
         LOCATE 19, 15: PRINT "(8) Enter the drive you want the
exported file on,"
         LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
1154      DN$ = INKEY$: IF DN$ = "" THEN 1154
1156      IN$ = DN$ + ":" + NF$ + ".txt"
```

```
        GOTO 1140
      ELSEIF DV$ = "9" THEN
        CLS
        LOCATE 21, 15: PRINT "(9) Enter the name of the data file,"
        LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
        LOCATE 23, 15: PRINT "    (default is Norm1):"
1158     INPUT NF$
1160     IN$ = DN$ + ":" + NF$ + ".txt"
        GOTO 1140
      ELSE
        GOTO 1150
      END IF
   ELSEIF D$ = "N" THEN
      GOTO 1170
   ELSEIF D$ = "Q" THEN
      GOTO 5000
   ELSE
      GOTO 1152
    END IF
1170 OPEN IN$ FOR OUTPUT AS #3
PRINT #3, "<<normeqd"
PRINT #3, "sigma=", sig
PRINT #3, "n=", n
PRINT #3, "k=", k
PRINT #3, "t=", t
PRINT #3, "pr=", PCS
  IF DS$ = "N" THEN
     PRINT #3, "<<dest"
  ELSE
     PRINT #3, "fst[t,k]=", fst
     PRINT #3, "sec[t,k]=", sec
  END IF
PRINT #3, "R[t,k,pr]//N"
CLOSE #3
GOTO 5000
1200 CLS
LOCATE 6, 15:  PRINT "Normal-Ranking Means-Equal Sample Sizes,
 Known Variance-Solve for Sigma"
LOCATE 7, 15:  PRINT ""
LOCATE 8, 15: PRINT "(Input value and hit the ENTER key)"
LOCATE 9, 15: PRINT ""
INPUT "(1) Enter indifference parameter, delta"; del
```

```
INPUT "(2) Enter sample size"; n
INPUT "(3) Enter number of populations to be ranked"; k
INPUT "(4) Enter number of best populations desired"; t
INPUT "(5) Enter desired probability of correct selection"; PCS
LOCATE 15, 10: PRINT "Do you want to enter your own search values
 for sigma (Y or N)?"
LOCATE 16, 10: PRINT "(If 'N' then the program computes values for
 the search)"
1204  DS$ = INKEY$: IF DS$ = "" THEN GOTO 1204
     IF DS$ = "Y" THEN
     INPUT "(6) Enter first search value for sigma";
 fst
     INPUT "(7) Enter second search value for sigma";
 sec
     ELSEIF DS$ = "N" THEN
     GOTO 1205
     ELSE
     GOTO 1204
     END IF
1205 LOCATE 19, 15: PRINT "(8) Enter the drive you want the
 exported file on,"
LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
1210 INPUT DN$
   IF DN$ = "" THEN
      DN$ = "C"
   END IF
LOCATE 21, 15: PRINT "(9) Enter the name of the data file,"
LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
LOCATE 23, 15: PRINT "    (default is norm1):"
1220 INPUT NF$
    IF NF$ = "" THEN
       NF$ = "norm1"
    END IF
1230 IN$ = DN$ + ":" + NF$ + ".txt"
1240 CLS
1241 LOCATE 6, 15: PRINT "The following are the values that you
 assigned:"
LOCATE 7, 15: PRINT "(1) delta="; del
LOCATE 8, 15: PRINT "(2) sample size="; n
LOCATE 9, 15: PRINT "(3) number of populations="; k
LOCATE 10, 15: PRINT "(4) number of best populations="; t
LOCATE 11, 15: PRINT "(5) desired PCS="; PCS
```

```
   IF DS$ = "Y" THEN
      LOCATE 12, 15: PRINT "(6) first search value for sigma="; fst
      LOCATE 13, 15: PRINT "(7) second search value for sigma="; sec
   ELSE
      GOTO 1242
   END IF
1242 LOCATE 14, 15: PRINT "(8) drive to store file="; DN$
LOCATE 15, 15: PRINT "(9) name of exported file="; NF$
LOCATE 16, 15: PRINT ""
LOCATE 17, 15: PRINT "Do you want to make any changes (Y or N
 or Q to Quit)?"
1250 D$ = INKEY$: IF D$ = "" THEN GOTO 1250
IF D$ = "Y" THEN
       LOCATE 18, 15: PRINT "Which value do you want to change?"
       LOCATE 19, 15: PRINT "(enter number in parenthesis)"
1252    DV$ = INKEY$: IF DV$ = "" THEN GOTO 1252
       IF DV$ = "1" THEN
         CLS
         INPUT "(1) Enter indifference parameter, delta"; del
         GOTO 1240
       ELSEIF DV$ = "2" THEN
         CLS
         INPUT "(2) Enter sample size, n"; n
         GOTO 1240
       ELSEIF DV$ = "3" THEN
         CLS
         INPUT "(3) Enter number of populations to be ranked"; k
         GOTO 1240
       ELSEIF DV$ = "4" THEN
         CLS
         INPUT "(4) Enter number of best populations desired"; t
         GOTO 1240
       ELSEIF DV$ = "5" THEN
         CLS
         INPUT "(5) Enter desired probability of correct selection";
 PCS
         GOTO 1240
       ELSEIF DV$ = "6" THEN
         CLS
         INPUT "(6) Enter first search value for sigma"; fst
         GOTO 1240
       ELSEIF DV$ = "7" THEN
```

```
          CLS
          INPUT "(7) Enter second search value for sigma"; sec
          GOTO 1240
        ELSEIF DV$ = "8" THEN
          CLS
          LOCATE 19, 15: PRINT "(8) Enter the drive you want the
 exported file on,"
          LOCATE 20, 15: PRINT "    either A, B, or C (default is C):"
1254       DN$ = INKEY$: IF DN$ = "" THEN 1254
1256       IN$ = DN$ + ":" + NF$ + ".txt"
          GOTO 1240
        ELSEIF DV$ = "9" THEN
          CLS
          LOCATE 21, 15: PRINT "(9) Enter the name of the data file,"
          LOCATE 22, 15: PRINT "    (.txt data extension assumed),"
          LOCATE 23, 15: PRINT "    (default is Norm1):"
1258       INPUT NF$
1260       IN$ = DN$ + ":" + NF$ + ".txt"
          GOTO 1240
        ELSE
          GOTO 1250
        END IF
    ELSEIF D$ = "N" THEN
       GOTO 1270
    ELSEIF D$ = "Q" THEN
       GOTO 5000
    ELSE
       GOTO 1252
     END IF
1270 OPEN IN$ FOR OUTPUT AS #4
PRINT #4, "<<normeqs"
PRINT #4, "delta=", del
PRINT #4, "n=", n
PRINT #4, "k=", k
PRINT #4, "t=", t
PRINT #4, "pr=", PCS
  IF DS$ = "N" THEN
     PRINT #4, "<<sest"
  ELSE
     PRINT #4, "fst[t,k]=", fst
     PRINT #4, "sec[t,k]=", sec
  END IF
```

```
PRINT #4, "R[t,k,pr]//N"
CLOSE #4
GOTO 5000
1300 CLS
REM
REM The following begins the Level III unequal sample size menu
REM options. At this time, the user can only choose one
REM population from k.
REM
LOCATE 6, 15: PRINT "Normal-Ranking Means-Unequal Sample Sizes,
 Known Variance"
LOCATE 7, 15: PRINT ""
LOCATE 8, 15: PRINT "(Select option.  You do not have to press
 ENTER)"
LOCATE 9, 15: PRINT ""
LOCATE 10, 15: PRINT "Options are:"
LOCATE 11, 15: PRINT ""
LOCATE 12, 15: PRINT "  (1) Number of best populations is 1"
LOCATE 13, 15: PRINT "* (2) Number of best populations is
more than 1"
LOCATE 14, 15: PRINT "  (3) Quit"
LOCATE 15, 15: PRINT ""
LOCATE 16, 15: PRINT "(*) indicates option is not available"
LOCATE 17, 15: PRINT ""
1309 LOCATE 18, 15: PRINT "Select Option:"
1310 D$ = INKEY$: IF D$ = "" THEN GOTO 1310
1320 IF D$ = "1" THEN
         LOCATE 20, 15: PRINT "You have selected the t=1 option."
         LOCATE 21, 15: PRINT "Is this what you want to do
(Y or N)?"
1330     D$ = INKEY$: IF D$ = "" THEN GOTO 1330
         IF D$ = "Y" THEN
           GOTO 1400
         ELSEIF D$ = "N" THEN
           GOTO 1300
         ELSE
           GOTO 1330
         END IF
     ELSEIF D$ = "2" THEN
         LOCATE 20, 15: PRINT "Option not available at this time"
         LOCATE 21, 15: PRINT "Reselect Option:"
         GOTO 1309
```

```
    ELSEIF D$ = "3" THEN
         LOCATE 20, 15: PRINT "You have selected to Quit."
         LOCATE 21, 15: PRINT "Is this what you want to do
(Y or N)?"
1340     D$ = INKEY$: IF D$ = "" THEN GOTO 1340
         IF D$ = "Y" THEN
           STOP
         ELSEIF D$ = "N" THEN
           GOTO 1300
         ELSE
           GOTO 1340
         END IF
    END IF
1400 CLS
REM
REM Level IV options for unequal sample sizes.  Can solve for
REM delta or the PCS.
REM
LOCATE 6, 15· PRINT "Normal-Ranking Means-Unequal Sample Size,
 Common V ·i nce"
LOCATE 7, 15: PRINT "1 Best Population"
LOCATE 7, 15: PRINT "; "; ""
LOCATE 8, 15: PRINT "(Select option.  You do not have to press
 ENTER)"
LOCATE 9, 15: PRINT ""
LOCATE 10, 15: PRINT "Options are:"
LOCATE 11, 15: PRINT ""
LOCATE 12, 15: PRINT "(1) Solving for PCS When Sample Sizes
are Known"
LOCATE 13, 15: PRINT "(2) Solving for Delta When Sample Sizes
are Known"
LOCATE 14, 15: PRINT "(3) Quit"
LOCATE 15, 15: PRINT ""
1409 LOCATE 16, 15: PRINT "Select Option:"
1410 D$ = INKEY$: IF D$ = "" THEN GOTO 1410
1412 IF D$ = "1" THEN
         LOCATE 20, 15: PRINT "You have selected to solve for
the PCS."
         LOCATE 21, 15: PRINT "Is this what you want to do
(Y or N)?"
1420     D$ = INKEY$: IF D$ = "" THEN GOTO 1420
         IF D$ = "Y" THEN
```

```
          GOTO 1500
         ELSEIF D$ = "N" THEN
          GOTO 1400
         ELSE
          GOTO 1420
         END IF
      ELSEIF D$ = "2" THEN
         LOCATE 20, 15: PRINT "You have selected to solve
 for delta"
         LOCATE 21, 15: PRINT "Is this what you want to do
(Y or N)?"
1430     D$ = INKEY$: IF D$ = "" THEN GOTO 1430
         IF D$ = "Y" THEN
          GOTO 1600
         ELSEIF D$ = "N" THEN
          GOTO 1400
         ELSE
          GOTO 1430
         END IF
      ELSEIF D$ = "3" THEN
         LOCATE 20, 15: PRINT "You have selected to Quit."
         LOCATE 21, 15: PRINT "Is this what you want to do
(Y or N)?"
1440     D$ = INKEY$: IF D$ = "" THEN GOTO 1440
         IF D$ = "Y" THEN
          STOP
         ELSEIF D$ = "N" THEN
          GOTO 1400
         ELSE
          GOTO 1440
         END IF
    END IF
1500 CLS
REM
REM Level V Menu option for unequal sample sizes, solving
REM for the PCS. The program asks the user input.
REM k sample sizes should be entered.
REM
LOCATE 6, 15: PRINT "Normal-Ranking Means-Unequal Sample Sizes,
 Known Variance"
LOCATE 7, 15: PRINT "1 Best Population-Solve for PCS"
LOCATE 8, 15: PRINT ""
```

```
LOCATE 9, 15: PRINT "(Input value and hit the ENTER key)"
LOCATE 10, 15: PRINT ""
INPUT "(1) Enter indifference paramter, delta"; del
INPUT "(2) Enter standard deviation, sigma"; sig
INPUT "(3) Enter number of populations to be ranked"; k
INPUT "(4) Enter first sample size"; SS(1)
Q = 2
DO
  INPUT "Enter next sample size"; SS(Q)
  Q = Q + 1
LOOP WHILE (Q < k + 1)
CLS
LOCATE 19, 15: PRINT "Enter the drive you want the exported
file on,"
LOCATE 20, 15: PRINT "either A, B, or C (default is C):"
1510 INPUT DN$
   IF DN$ = "" THEN
      DN$ = "C"
   END IF
LOCATE 21, 15: PRINT "Enter the name of the data file,"
LOCATE 22, 15: PRINT "(.txt data extension assumed),"
LOCATE 23, 15: PRINT "(default is norm1):"
1520 INPUT NF$
    IF NF$ = "" THEN
       NF$ = "norm1"
    END IF
1530 IN$ = DN$ + ":" + NF$ + ".txt"
OPEN IN$ FOR OUTPUT AS #5
REM
REM The program automatically arranges the sample size ordering
REM to produce k PCS values.  It does this by switching
REM all of the k-1 sample sizes with the kth sample size.
REM Defining m and n as tables allows the program
REM to keep track of this ordering and output the
REM ordering that corresponds to the PCS value.  The user
REM can look at the output to determine the
REM sample size order that corresponds to the lowest PCS.
REM
PRINT #5, "<<normueq"
PRINT #5, "delta=", del
PRINT #5, "sigma=", sig
PRINT #5, "k=", k
```

```
PRINT #5, "m=Table[x,{k}]"
PRINT #5, "n=Table[x,{k}]"
R = 1
SS(0) = SS(k)
DO
  Q = 1
  DO
    PRINT #5, "n[["; Q; "]] =", SS(Q)
    Q = Q + 1
  LOOP WHILE (Q < k + 1)
  PRINT #5, "Q[k]//N"
  PRINT #5, "m[["; R; "]]=N[%]"
  SWAP SS(k), SS(k - R + 1)
  SWAP SS(k), SS(k - R)
  R = R + 1
LOOP WHILE (R < k + 1)
PRINT #5, "Min[m]"
CLOSE #5
GOTO 5000
1600 CLS
REM
REM The following Level V Menu solves for the indifference
REM parameter for unequal sample sizes.  It follows a
REM similar procedure as the previous one.
REM
LOCATE 6, 15: PRINT "Normal-Ranking Means-Unequal Sample Size,
 Known Variance"
LOCATE 7, 15: PRINT "1 Best Population-Solve for Delta"
LOCATE 8, 15: PRINT ""
LOCATE 9, 15: PRINT "(Input value and hit the ENTER key)"
LOCATE 10, 15: PRINT ""
INPUT "(1) Enter the desired probability of correct selection, PCS";
PCS
INPUT "(2) Enter standard deviation, sigma"; sig
INPUT "(3) Enter number of populations to be ranked"; k
INPUT "(4) Enter first search value for delta"; fst
INPUT "(5) Enter second search value for delta"; sec
INPUT "(6) Enter first sample size"; SS(1)
Q = 2
REM The following statement sums the square roots of the
REM sample sizes for the GOS procedure.  This is an
REM addition to the program to have the software determine
```

```
REM search values for delta.  Currently the user has to
REM enter their own values into the computer.  If this
REM addition is eventually implemented, the above
REM input statements (4) and (5) can be remarked out or
REM the user can be provided an option of whether to
REM enter their own values or rely on the computer,
REM similar to the equal sample size options.
REM
SM = SQR(SS(1))
DO
  INPUT "Enter next sample size"; SS(Q)
REM Want to sum the the square roots of the sample sizes to get an
REM estimate for n, an average sample size.  Then we can use the
REM GOS approximation and determine search values for delta,
REM using the fitted regression equation obtained from
REM Bechhofer's Table (see Appendix D).
REM
SM = SQR(SS(Q)) + SM
  Q = Q + 1
LOOP WHILE (Q < k + 1)
CLS
LOCATE 19, 15: PRINT "Enter the drive you want the exported
 file on,"
LOCATE 20, 15: PRINT "either A, B, or C (default is C):"
1610 INPUT DN$
   IF DN$ = "" THEN
      DN$ = "C"
   END IF
LOCATE 21, 15: PRINT "Enter the name of the data file,"
LOCATE 22, 15: PRINT "(.txt data extension assumed),"
LOCATE 23, 15: PRINT "(default is norm1):"
1620 INPUT NF$
    IF NF$ = "" THEN
       NF$ = "norm1"
    END IF
1630 IN$ = DN$ + ":" + NF$ + ".txt"
OPEN IN$ FOR OUTPUT AS #6
REM The following remarked out statements were added to the
REM program.  This addition allows the delta search
REM values to be estimated by the same fitted equation
REM that was previously used in the equal sample
REM size computation files.  Here, the equations are the
```

```
REM same, only implemented in QuickBASIC instead of
REM Mathematica.  The fitted equation can be used since
REM an average n is calculated using the GOS procedure.
REM This should provide two reasonable values to
REM initiate the search for delta.
REM
REM NO = (SM / k)
REM EP = EXP(ATN(PCS / SQR(1 - (PCS) ^ 2)))
REM TV1 = -3.12497 + .34799 * k + .42398 - .01989 * k ^ 2
REM TV2 = -.10058 + .04482 * k + 1.43618 * EP + .00108 * k ^ 2 * EP
REM TV3 = -.01819 * k * EP - .00462 * k * EP + .01263 * EP -
  .06606 * EP
REM TV = TV1 + TV2 + TV3
REM fst = TV * sig / NO
REM sec = 1.1 * fst
PRINT #6, "<<normueqd"
PRINT #6, "Q[k]="; PCS
PRINT #6, "sigma="; sig
PRINT #6, "k="; k
PRINT #6, "m=Table[x,{k}]"
PRINT #6, "n=Table[x,{k}]"
R = 1
SS(0) = SS(k)
DO
  Q = 1
  DO
    PRINT #6, "n[["; Q; "]] =", SS(Q)
    Q = Q + 1
  LOOP WHILE (Q < k + 1)
  PRINT #6, "D["; k; ","; PCS; ","; fst; ","; sec; "]//N"
  PRINT #6, "m[["; R; "]]=N[%]"
  SWAP SS(k), SS(k - R + 1)
  SWAP SS(k), SS(k - R)
  R = R + 1
LOOP WHILE (R < k + 1)
PRINT #6, "Min[m]"
CLOSE #6
5000 END
```

Appendix D. Mathematica Computational Files

This appendix contains files created in *Mathematica* to calculate the indifference-zone integral expression and the search values needed by *Mathematica*'s root finding function.

D.1 Files Used To Calculate the Indifference-Zone Integral Expression for Various Parameters

The QuickBASIC menu program calls upon one of these files, depending on the parameter that is unknown to the experimenter. These *Mathematica* statements are based on the integral expressions developed in Chapter 2.

```
<<Statistics'ContinuousDistributions'
F[t_]:=CDF[NormalDistribution[0,1],t]
G[p_]:=Quantile[NormalDistribution[0,1],p]
Q[n_,t_,k_]:=t*NIntegrate[(F[G[u]+(delta/sigma)*Sqrt[n]])^(k-t)*
(1-u)^(t-1),{u,0,1}]
R[t_,k_,pr_]:=FindRoot[Q[n,t,k]==pr,{n,fst[t,k],sec[t,k]}]
```

Figure D.1 'Normeq' *Mathematica* File Used To Solve For the PCS or n For Normal Populations of Equal Sample Size.

```
<<Statistics'ContinuousDistributions'
F[t_]:=CDF[NormalDistribution[0,1],t]
G[p_]:=Quantile[NormalDistribution[0,1],p]
Q[delta_,t_,k_]:=t*NIntegrate[(F[G[u]+(delta/sigma)*Sqrt[n]])^(k-t)*
(1-u)^(t-1),{u,0,1}]
R[t_,k_,pr_]:=FindRoot[Q[delta,t,k]==pr,{delta,fst[t,k],sec[t,k]}]
```

Figure D.2 'Normeqd' *Mathematica* File Used To Solve For the Indifference Parameter For Normal Populations of Equal Sample Size.

```
<<Statistics`ContinuousDistributions`
F[t_]:=CDF[NormalDistribution[0,1],t]
G[p_]:=Quantile[NormalDistribution[0,1],p]
Q[sigma_,t_,k_]:=t*NIntegrate[(F[G[u]+(delta/sigma)*Sqrt[n]])^(k-t)*
(1-u)^(t-1),{u,0,1}]
R[t_,k_,pr_]:=FindRoot[Q[sigma,t,k]==pr,{sigma,fst[t,k],sec[t,k]}]
```

Figure D.3 'Normeqs' *Mathematica* File Used To Solve For the Standard Deviation For Normal Populations of Equal Sample Size.

```
<<Statistics`ContinuousDistributions`
F[t_]:=CDF[NormalDistribution[0,1],t]
G[p_]:=Quantile[NormalDistribution[0,1],p]
Q[k_]:=NIntegrate[NProduct[F[(Sqrt[n[[i]]/n[[k]]])*G[u]+
(Sqrt[n[[i]]]/sigma)*delta],{i,1,k-1}],{u,0,1}]
```

Figure D.4 'Normueq' *Mathematica* File Used To Solve For the PCS For Normal Populations of Unequal Sample Size.

```
<<Statistics`ContinuousDistributions`
F[t_]:=CDF[NormalDistribution[0,1],t]
G[p_]:=Quantile[NormalDistribution[0,1],p]
Q[k_,delta_]:=NIntegrate[NProduct[F[(Sqrt[n[[i]]/n[[k]]])*G[u]+
(Sqrt[n[[i]]]/sigma)*delta],{i,1,k-1}],{u,0,1}]
D[k_,pr_,fst_,sec_]:=FindRoot[Q[k,delta]==pr,{delta,fst,sec}]
```

Figure D.5 'Normueqd' *Mathematica* File Used To Solve For Indifference Parameter for Normal Populations of Unequal Sample Size.

D.2 *Files Used To Estimate Search Values For Mathematica's Root Finding Function*

The files listed in Figures D-6, D-7, and D-8 automatically compute the search values needed to determine n, delta, and sigma, respectively, for the equal sample size case. These search values are computed using

$$\begin{aligned} TV &= -3.12497 + 0.34799k + 0.42398t - 0.01989k^2 - 0.10058t^2 + 0.04482kt \\ &\quad +1.43618p + 0.00108k^2p - 0.01819kp - 0.00462ktp + 0.01263t^2p \\ &\quad -0.06606tp \end{aligned}$$

where $TV = \delta\sqrt{n}/\sigma$ is the tabulated value from Bechhofer's table,

$k =$ the number of populations under consideration,

$t =$ the number of populations to be selected, and

$p = \exp(\arcsin(pr))$ is a transformation of the PCS.

The expression for TV is a least squares approximation of 500 values arbitrarily chosen from Bechhofer's table [1] [3:30-36]. The approximating equation has a fairly high coefficient of determination (R^2 =0.9800, adjusted R^2 =0.9796) and generally predicts values within ten percent of those in the table.

The predicted value for TV is used to approximate the search values for n, δ, or σ through simple algebraic manipulation. For example, $n = (TV\sigma/\delta)^2$ would be the first search parameter needed to solve the integral expression for n. The second search parameter is a factor of 1.1 times the first.

[1] Dr. David Barr produced the expression using STATISTIX, a statistical computer software package.

```
p[pr_]:=Exp[ArcSin[pr]]
TV[t_,k_]:=-3.12497+.3479*k+.42398*t-(.01989*(k^2))-(.10058*(t^2))+
(.04482*k*t)+(1.43618*p[pr])+(.00108(k^2)*p[pr])-(.01819*k*p[pr])-
(.00462*k*t*p[pr])+(.01263*(t^2)*p[pr])-(.06606*t*p[pr])
fst[t_,k_]:=(TV[t,k]*sigma/delta)^2
sec[t_,k_]:=.10*fst[t,k]+fst[t,k]
```

Figure D.6 'Nest' *Mathematica* File Used To Estimate Two Search Values for n, the Common Sample Size.

```
p[pr_]:=Exp[ArcSin[pr]]
TV[t_,k_]:=-3.12497+.3479*k+.42398*t-(.01989*(k^2))-(.10058*(t^2))+
(.04482*k*t)+(1.43618*p[pr])+(.00108(k^2)*p[pr])-(.01819*k*p[pr])-
(.00462*k*t*p[pr])+(.01263*(t^2)*p[pr])-(.06606*t*p[pr])
fst[t_,k_]:=(TV[t,k]*sigma/Sqrt[n])
sec[t_,k_]:=.10*fst[t,k]+fst[t,k]
```

Figure D.7 'Dest' *Mathematica* File Used To Estimate Two Search Values for δ, the Indifference Parameter.

```
p[pr_]:=Exp[ArcSin[pr]]
TV[t_,k_]:=-3.12497+.3479*k+.42398*t-(.01989*(k^2))-(.10058*(t^2))+
(.04482*k*t)+(1.43618*p[pr])+(.00108(k^2)*p[pr])-(.01819*k*p[pr])-
(.00462*k*t*p[pr])+(.01263*(t^2)*p[pr])-(.06606*t*p[pr])
fst[t_,k_]:=(Sqrt[n]*delta/TV[t,k])
sec[t_,k_]:=.10*fst[t,k]+fst[t,k]
```

Figure D.8 'Sest' *Mathematica* File Used To Estimate Two Search Values for σ, the Standard Deviation.

Appendix E. Examples of QuickBASIC Output Files For Input Into Mathematica

The following figures are examples of the files that are created by the QuickBASIC computer program. Each figure represents an example for each of the Level IV (for equal sample sizes) and Level V (for unequal sample sizes) menu options. These files invoke the *Mathematica* computational files displayed in Appendix D. The user can provide a unique name to each file for input into *Mathematica*.

```
<<normeq
delta =        4
sigma =        10
k=             5
t=             2
pr=            .95
<<nest
R[t,k,pr]//N
```

Figure E.1 Example of a QuickBASIC Output File For the Specific Case of Normally Distributed Populations With Equal Sample Size and Solving for n; Given k =5, t =2, PCS=.95, δ =4, σ =10 and the search values for n determined by a *Mathematica* file, 'nest'.

```
<<normeq
delta=        4
sigma=        10
k=            5
t=            2
n=            15
Q[n,t,k]//N
```

Figure E.2 Example of a QuickBASIC Output File For the Specific Case of Normally Distributed Populations With Equal Sample Size and Solving for the PCS; Given k =5, t =2, n =15, δ =4, and σ =10.

```
<<normeqd
sigma=        10
n=            15
k=             5
t=             2
pr=           .95
<<dest
R[t,k,pr]//N
```

Figure E.3 Example of a QuickBASIC Output File For the Specific Case of Normally Distributed Populations With Equal Sample Size and Solving for δ; Given k =5, t =2, PCS=.95, n = 15, σ =10 and search values determined by a *Mathematica* file, 'dest'.

```
<<normeqs
delta=        4
n=            15
k=            5
t=            2
pr=           .95
<<sest
R[t,k,pr]//N
```

Figure E.4 Example of a QuickBASIC Output File For the Specific Case of Normally Distributed Populations With Equal Sample Size and Solving for σ; Given k =5, t =2, PCS=.95, n = 15, δ =4 and search values determined by a *Mathematica* file, 'sest'.

```
<<normueq
delta=        4
sigma=        10
k=            5
m=Table[x,{k}]
n=Table[x,{k}]
n[[ 1 ]] =    15
n[[ 2 ]] =    15
n[[ 3 ]] =    15
n[[ 4 ]] =    14
n[[ 5 ]] =    14
Q[k]//N
m[[ 1 ]]=N[%]
n[[ 1 ]] =    15
n[[ 2 ]] =    15
n[[ 3 ]] =    15
n[[ 4 ]] =    14
n[[ 5 ]] =    14
Q[k]//N
m[[ 2 ]]=N[%]
n[[ 1 ]] =    15
n[[ 2 ]] =    15
n[[ 3 ]] =    14
n[[ 4 ]] =    14
n[[ 5 ]] =    15
Q[k]//N
m[[ 3 ]]=N[%]
n[[ 1 ]] =    15
n[[ 2 ]] =    14
n[[ 3 ]] =    15
n[[ 4 ]] =    14
n[[ 5 ]] =    15
Q[k]//N
m[[ 4 ]]=N[%]
n[[ 1 ]] =    14
n[[ 2 ]] =    15
n[[ 3 ]] =    15
n[[ 4 ]] =    14
n[[ 5 ]] =    15
Q[k]//N
m[[ 5 ]]=N[%]
Min[m]
```

Figure E.5 Example of a QuickBASIC Output File For the Specific Case of Normally Distributed Populations With Unequal Sample Sizes and Solving for the PCS; Given k =5, t =1, δ =4, σ =10, and sample sizes of 15, 15, 15, 14, and 14.

```
<<normueqd
Q[k]= .95
sigma= 10
k= 5
m=Table[x,{k}]
n=Table[x,{k}]
n[[ 1 ]] =     15
n[[ 2 ]] =     15
n[[ 3 ]] =     15
n[[ 4 ]] =     14
n[[ 5 ]] =     14
D[ 5 , .95 , 5 , 20 ]//N
m[[ 1 ]]=N[%]
n[[ 1 ]] =     15
n[[ 2 ]] =     15
n[[ 3 ]] =     15
n[[ 4 ]] =     14
n[[ 5 ]] =     14
D[ 5 , .95 , 5 , 20 ]//N
m[[ 2 ]]=N[%]
n[[ 1 ]] =     15
n[[ 2 ]] =     15
n[[ 3 ]] =     14
n[[ 4 ]] =     14
n[[ 5 ]] =     15
D[ 5 , .95 , 5 , 20 ]//N
m[[ 3 ]]=N[%]
n[[ 1 ]] =     15
n[[ 2 ]] =     14
n[[ 3 ]] =     15
n[[ 4 ]] =     14
n[[ 5 ]] =     15
D[ 5 , .95 , 5 , 20 ]//N
m[[ 4 ]]=N[%]
n[[ 1 ]] =     14
n[[ 2 ]] =     15
n[[ 3 ]] =     15
n[[ 4 ]] =     14
n[[ 5 ]] =     15
D[ 5 , .95 , 5 , 20 ]//N
m[[ 5 ]]=N[%]
Min[m]
```

Figure E.6 Example of a QuickBASIC Output File For the Specific Case of Normally Distributed Populations With Unequal Sample Sizes and Solving for δ; Given k =5, t =1, PCS of .95, σ =10, and sample sizes of 15, 15, 15, 14, and 14.

Appendix F. Indifference-Zone Integral Expression For Normally Distributed Populations With Unequal Sample Sizes For the Case of k =2 and t =1, Written in MATHCAD

The following is the MATHCAD formula used to compute the values for Table 4.3. It shows a specific example using $n_{(1)}$ =57, $n_{(2)}$ =60, and δ/σ =.4. MATHCAD has a 15 digit accuracy.

$$n1 := 57$$

$$n2 := 60$$

$$q(n1, n2) := \int_0^1 \mathrm{cnorm}\left(\sqrt{\frac{n1}{n2}} \cdot \mathrm{root}(\mathrm{cnorm}(x) - u, x) + .4 \cdot \sqrt{n1}\right) du$$

$$q(n1, n2) = 0.984716550593042$$

Appendix G. PCS vs. λ Graphs for Normally Distributed Populations With Unequal Sample Sizes For the Case of $k = 2$ and $t = 1$.

The following are graphs of Equation (4.9) and (4.10) at varying δ/σ ratios;

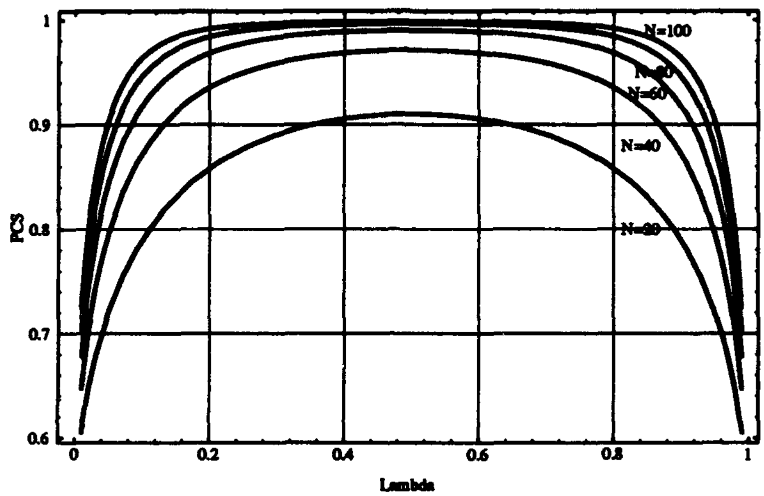

Figure G.1 PCS vs. λ resulting from Equation (4.10) For Various N When δ/σ =.6.

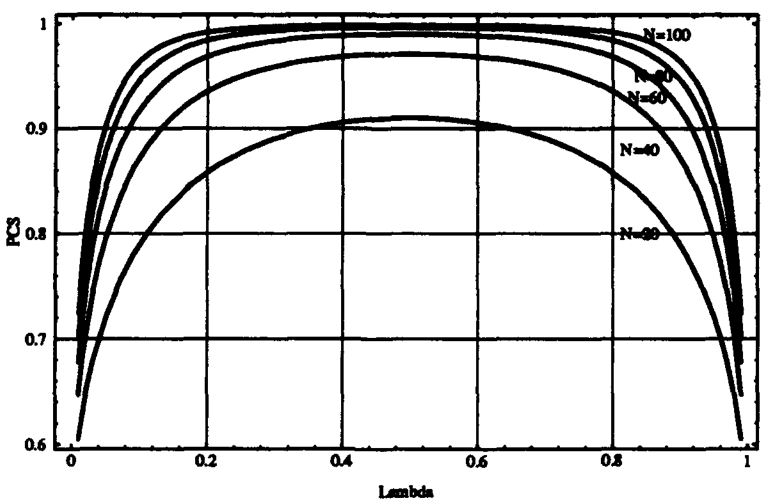

Figure G.2 PCS vs. λ resulting from Equation (4.11) For Various N When δ/σ =.6.

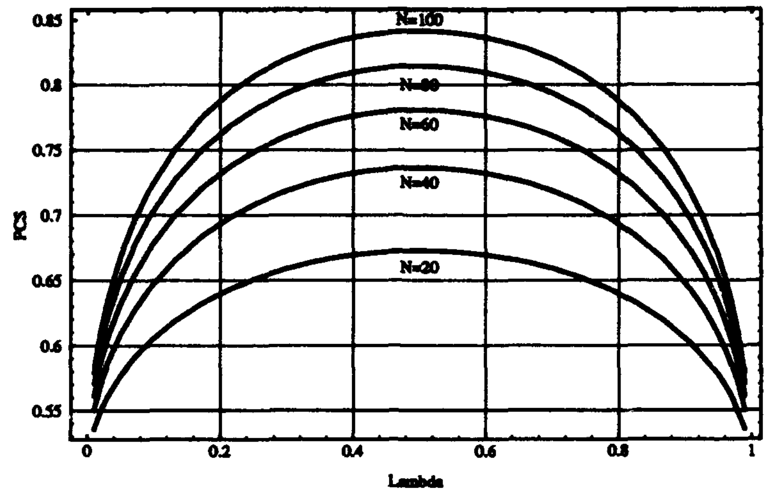

Figure G.3 PCS vs. λ resulting from Equation (4.10) For Various N When $\delta/\sigma = .2$.

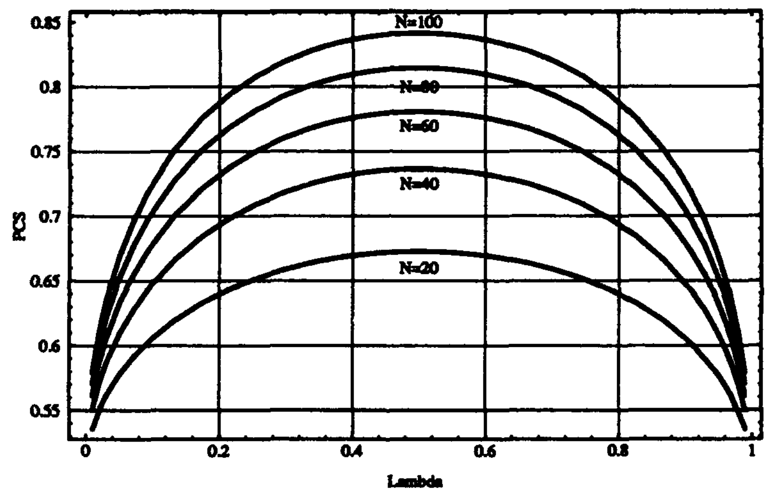

Figure G.4 PCS vs. λ resulting from Equation (4.11) For Various N When $\delta/\sigma = .2$.

Appendix H. Data Tables

Table H.1 PCS Values as a Result of Losing One Observation From Each of Two Samples and Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =4, t =2. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	*PCS*
14	14	15	15	0.637850
14	15	14	15	0.637874
14	15	15	14	0.637874
15	14	14	15	0.637874
15	14	15	14	0.637874
15	15	14	14	0.637841

Table H.2 PCS Values as a Result of Losing One Observation From Each of Two Samples and Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =6, t =3. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	$n_{(6)}$	*PCS*
14	14	15	15	15	15	0.449046
14	15	14	15	15	15	0.449046
15	14	14	15	15	15	0.449046
15	15	15	14	14	15	0.449046
15	15	15	14	15	14	0.449046
15	15	15	15	14	14	0.449046
14	15	15	14	15	15	0.449054
14	15	15	15	14	15	0.449054
14	15	15	15	15	14	0.449054
15	14	15	14	15	15	0.449054
15	14	15	15	14	15	0.449054
15	14	15	15	15	14	0.449054
15	15	14	14	15	15	0.449054
15	15	14	15	14	15	0.449054
15	15	14	15	15	14	0.449054

Table H.3 PCS Values as a Result of Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =4, t =2. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	PCS
15	15	15	15	0.646328
14	15	15	15	0.642070
15	14	15	15	0.642070
15	15	14	15	0.642070
15	15	15	14	0.642070
13	15	15	15	0.637399
15	13	15	15	0.637399
15	15	13	15	0.637399
15	15	15	13	0.637399

Table H.4 PCS Values as a Result of Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =4, t =3. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	PCS
15	15	15	15	0.715280
15	14	15	15	0.711844
15	15	14	15	0.711844
15	15	15	14	0.711844
14	15	15	15	0.710928
15	14	14	15	0.708432
15	14	15	14	0.708432
15	15	14	14	0.708432
14	14	15	15	0.707575
14	15	14	15	0.707575
14	15	15	14	0.707575
15	13	15	15	0.707983
15	15	13	15	0.707983
15	15	15	13	0.707983
13	15	15	15	0.706216

Table H.5 PCS Values as a Result of Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =5, t =1. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	PCS
15	15	15	15	15	0.667542
14	15	15	15	15	0.664453
15	14	15	15	15	0.664453
15	15	14	15	15	0.664453
15	15	15	14	15	0.664453
15	15	15	15	14	0.663790
14	14	15	15	15	0.661385
14	15	14	15	15	0.661385
14	15	15	14	15	0.661385
15	14	14	15	15	0.661385
15	14	15	14	15	0.661385
15	15	14	14	15	0.661385
14	15	15	15	14	0.660776
15	14	15	15	14	0.660776
15	15	14	15	14	0.660776
15	15	15	14	14	0.660776
13	15	15	15	15	0.660967
15	13	15	15	15	0.660967
15	15	13	15	15	0.660967
15	15	15	13	15	0.660967
15	15	15	15	13	0.659756

Table H.6 PCS Values as a Result of Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =5, t =3. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	PCS
15	15	15	15	15	0.557595
14	15	15	15	15	0.553634
15	14	15	15	15	0.553634
15	15	14	15	15	0.553877
15	15	15	14	15	0.553877
15	15	15	15	14	0.553877
15	15	14	14	15	0.550191
15	15	14	15	14	0.550191
15	15	15	14	14	0.550191
14	15	14	15	15	0.549966
14	15	15	14	15	0.549966
14	15	15	15	14	0.549966
15	14	14	15	15	0.549966
15	14	15	14	15	0.549966
15	14	15	15	14	0.549966
14	14	15	15	15	0.549713
15	15	13	15	15	0.549786
15	15	15	13	15	0.549786
15	15	15	15	13	0.549786
13	15	15	15	15	**
13	15	15	15	15	**

** Computer software could not provide a numerical answer for an unknown reason.

Table H.7 PCS Values as a Result of Varying Sample Size Assignments for Unequal Sample Size Populations in the Case of k =5, t =4. Note: δ/σ =.4 For All Cases.

$n_{(1)}$	$n_{(2)}$	$n_{(3)}$	$n_{(4)}$	$n_{(5)}$	PCS
15	15	15	15	15	0.667542
14	15	15	15	15	0.663790
15	14	15	15	15	0.664453
15	15	14	15	15	0.664453
15	15	15	14	15	0.664453
15	15	15	15	14	0.664453
15	14	14	15	15	0.661385
15	14	15	14	15	0.661385
15	14	15	15	14	0.631385
15	15	14	14	15	0.631385
15	15	14	15	14	0.631385
15	15	15	14	14	0.631385
14	14	15	15	15	0.660776
14	15	14	15	15	0.660776
14	15	15	14	15	0.660776
14	15	15	15	14	0.660776
15	13	15	15	15	0.660967
15	15	13	15	15	0.660967
15	15	15	13	15	0.660967
15	15	15	15	13	0.660967
13	15	15	15	15	0.659756

Bibliography

1. Barr, David R. and M. Haseeb Rizvi. "An Introduction to Ranking and Selection Procedures," *Journal of the American Statistical Association*, *61*:640 - 646 (1966).
2. Barr, David R. and M. Haseeb Rizvi. "Ranking and Selection Problems of Uniform Distributions," *Trabajos De Estadistica*, 15 - 31 (1966).
3. Bechhofer, Robert E. "A Single-Sample Multiple Decision Procedure For Ranking Means of NormalPopulations With Known Variances," *Annals of Mathematical Statistics*, *25*:16 - 39 (1954).
4. Gibbons, Jean D., Ingram Olkin and Milton Sobel. *Selecting and Ordering Populations: A New Statistical Methodology*. New York: John Wiley and Sons Inc., 1977.
5. Hari, V.V., Murat Tanik and Udo W. Pooch. *Illustrated QuickBASIC 4.0*. Plano, Texas: Wordware Publishing, Inc., 1989.
6. Mendenhall, William, Dennis D. Wackerly and Richard L. Scheaffer. *Mathematical Statistics With Applications*. Boston, Massachusetts: PWS-Kent Publishing Company, 1990.
7. Ross, Sheldon. *A First Course In Probability*. New York: Macmillan Publishing Company, 1988.
8. Wolfram, Stephen. *Mathematica; A System for Doing Mathematics by Computer*. Redwood City, California: Addison-Wesley Publishing Company, 1991.

Vita

Captain Catherine Annette Poston was born on September 9, 1966 in Cheverly, Maryland. She was a graduate of Eleanor Roosevelt High School in Greenbelt, Maryland in June of 1984. In September of 1988, she was a graduate of Michigan State University with a Bachelor of Science degree in Mechanical Engineering. She was a Reserve Officers Training Corps (ROTC) Distinguished Graduate and was awarded a regular commission upon graduation. In March of 1989, Captain Poston was a graduate of Undergraduate Space Training, Lowry AFB, Colorado. After three years of launch and early orbit operations for the NAVSTAR Global Positioning System (GPS) at the 1st Space Operations Squadron, 50th Space Wing, AFSPACECOM, Falcon AFB, Colorado, Captain Poston entered the Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Permanent address: 5506 Hamilton Street
Hyattsville, MD 20781

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
December 1993

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
SOLVING THE RANKING AND SELECTION INDIFFERENCE-ZONE FORMULATION FOR NORMAL DISTRIBUTIONS USING COMPUTER SOFTWARE

5. FUNDING NUMBERS

6. AUTHOR(S)
Catherine A. Poston, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSO/ENS/ENC/93D-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
Ranking and selection procedures are statistical methods used to compare and choose the best among a group of similar statistically distributed populations. The two predominant approaches to solving ranking and selection problems are Gupta's subset selection formulation and Bechhofer's indifference-zone formulation. For the indifference-zone formulation where the populations have equal sample sizes, Barr and Risvi developed an integral expression of the probability of correct selection (PCS). Given appropriate parameters, the integral expression can be solved to determine the common sample size required to attain a desired PCS. Tables with selected solutions to the integral expression are available for a variety of population distributions. These tables, however, are not included in any single reference, sometimes require interpolation, and only provide approximate results for the case of unequal sample sizes. Using a computer software program to solve the integral expression for the unknown parameters can eliminate these burdens. This paper describes the computer software developed to solve the integral expression of the indifference-zone formulation for normally distributed populations having either equal or unequal sample sizes. The software was written in QuickBASIC and *Mathematica*. The QuickBASIC code is a menu-driven interface that develops input files for *Mathematica*. *Mathematica* is the mathematical software package which performs the computationally intensive calculations required to solve the integral expressions. It is hoped that other ranking and selection problems will eventually apply this easy-to-use interface, making the ranking and selection procedure a more common tool in statistical decision making.

14. SUBJECT TERMS
Ranking and Selection, Ranking and Selection Procedures, Indifference-Zone, Indifference-Zone Formulation

15. NUMBER OF PAGES
131

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL