NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SEQUENTIAL MULTIPLE COMPARISON TESTING FOR BUDGET-LIMITED APPLICATIONS

by

Ofer Gonen

December 2004

Thesis Advisor:	Robert A. Koyak
Second Reader:	David Annis

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Sequential Multiple Comparison Testing for Budget-Limited Applications			**5. FUNDING NUMBERS**
6. AUTHOR(S) Major Ofer Gonen			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Computer simulations which forecast the performance of complicated systems are used as decision aids in many applications. For example, a ship's defensive system may use simulation to support an automated real-time response to a perceived threat, such as an incoming missile. The system uses cumulative simulation data to evaluate a set of options in order to choose the best countermeasure. Since everything happens in "real time", the system has limited time to run the simulation.

Normally, a system would run the simulation an equal number of times for each option before coming to a decision. But this may cause the system to waste time on options which can be deemed non-optimal after only a few simulation runs. This time can be better used to help adjudicate between the better options.

We evaluate the performance of sequential multiple comparisons algorithms to eliminate inferior options as quickly as possible, in order to have more time to dedicate to the exploration of better options, so that better decisions may be made. These algorithms allow inferior options to be dropped quickly depending on how well separated they are from others, but the algorithms differ in how well they achieve this objective.

14. SUBJECT TERMS Simulation, sequential analysis, multiple comparisons			**15. NUMBER OF PAGES** 73
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

SEQUENTIAL MULTIPLE COMPARISON TESTING FOR BUDGET-LIMITED APPLICATIONS

Ofer Gonen
Major, Israeli Air Force
B.S., Hebrew University of Jerusalem, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2004

Author: Ofer Gonen

Approved by: Robert A. Koyak
Thesis Advisor

David Annis
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Computer simulations which forecast the performance of complicated systems are used as decision aids in many applications. For example, a ship's defensive system may use simulation to support an automated real-time response to a perceived threat, such as an incoming missile. The system uses cumulative simulation data to evaluate a set of options in order to choose the best countermeasure. Since everything happens in "real time", the system has limited time to run the simulation.

Normally, a system would run the simulation an equal number of times for each option before coming to a decision. But this may cause the system to waste time on options which can be deemed non-optimal after only a few simulation runs. This time can be better used to help adjudicate between the better options.

We evaluate the performance of sequential multiple comparisons algorithms to eliminate inferior options as quickly as possible, in order to have more time to dedicate to the exploration of better options, so that better decisions may be made. These algorithms allow inferior options to be dropped quickly depending on how well separated they are from others, but the algorithms differ in how well they achieve this objective.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENT

This thesis would not have been possible without the assistance of Assistant Professor of Operations Research Robert A. Koyak, who agreed to be my advisor on this project even though it is not his field of expertise. I wish to thank him for the extra time and effort he invested in developing this topic with me.

I would also like to thank Ittai Avital for suggesting this thesis topic.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Computer simulations are often used as decision aids to forecast the performance of complicated systems. For example, a ship's defensive system may use simulation to support an automated real-time response to a perceived threat, such as an incoming missile. The ship has several countermeasures (options) from which to choose to respond to the threat. The defensive system wants to choose the option that is best; namely, that which minimizes the probability that the missile will hit the ship. Typically, the "kill probabilities" for the different options are unknown, because they depend on factors such as the number of missiles that were launched and their trajectories. To choose the best option the system uses available information about the threat, and conducts simulations to estimate the kill probabilities. These estimates are subject to uncertainty, which is decreased as the number of simulations is increased. But the threat must be engaged in real time. This limits the number of simulations that can be conducted while leaving enough time to successfully engage the threat.

We consider the problem of selecting the best from a set of options based on accumulating information, where the resources that can be devoted to simulation are limited. A possible solution to this problem is to run the simulation an equal number of times for each option before coming to a decision. But by doing so the system may waste time on options which can be deemed non-optimal after only a few simulation runs. This time can be better used to help adjudicate between the better options. All selection procedures that pre-allocate simulation resources among a set of options have this shortcoming.

Our objective is to identify selection procedures that allow the system to eliminate inferior options as quickly as possible, leaving more time to dedicate to the exploration of better options, and leading, hopefully, to better decisions. The selection procedures that we consider are designed to operate sequentially as simulation data accumulate in real time. By dropping clearly inferior options early, more resources can be dedicated to sharpening estimates of differences between options that are less clearly distinguished.

This is in contrast to fixed-allocation schemes, which are unable to dynamically allocate resources.

We examine a class of statistical techniques known as Sequential Multiple Comparisons with the Best (SMCB) to satisfy our objective. Although SMCB procedures have the sequential decision-making feature that we seek, they tend to be conservative in eliminating inferior options. This tendency arises from their formulation, which is structured to allow for the elimination of the best option with only a small probability. In our context, however, only one option is selected once the simulated budget is exhausted, regardless of how many options are "alive" at that point. Even if the best option has survived until the end, which is regarded as a "success" in the SMCB formulation, it may not survive our final selection, which we regard as a "failure".

At each sampling stage, a SMCB procedure maintains a set of options that is offered to contain the best option. Each option in this set is sampled once, and estimates of the means and variances of the options are updated. An elimination rule is then applied to these "live" options, and the set is updated. The set of live options is strictly non-increasing, and it may fail to converge to a single (putatively best) option once the procedure is terminated. Similarly, it may discard the best option erroneously at any stage. Although erroneous elimination risk is controlled by fixing the Type I error of the process, this and other aspects of SMCB procedures are calibrated to a set of distributional assumptions (normality, known or equal variances) that may be violated in practice.

For these reasons, it is not clear that SMCB procedures always perform better than non-sequential procedures. To examine the tradeoffs between the two types of procedures we conduct a series of simulation experiments in which several different types of SMCB procedures are compared to a "base case" of equal allocation of sampling effort across all options. We vary the number of options, their pattern of separation, and the probability distributions of the simulated observations. We also consider the effects of applying a SMCB procedure when its assumptions are violated. Comparisons are made using two measures of performance: probability of selecting the best option, and average difference of the means of the chosen and best options.

The primary benefit of using a SMCB procedure is to eliminate clearly inferior options early. If the best option is well separated from some of the others it is possible for a system to capitalize on this property. But this advantage is quickly lost if the options are not well separated or if important assumptions are violated. In the former case SMCB procedures are almost indistinguishable from non-sequential procedures unless the sampling budget is set very high. This is particularly true of SMCB procedures based on an assumption of normal data and unequal, unknown variances. SMCB procedures for normal data and equal, known variances can eliminate inferior options faster, but they carry a greater risk of eliminating the best option if the variance is specified incorrectly.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The problem's origins come from an anti-missile decision aid system. We do not get into details about this system, but we describe its important features, only to help the reader visualize and understand the problem.

The anti-missile system is activated as soon as an approaching missile is detected. The system receives different parameters such as the location and velocity of the incoming missile, the wind state and the time of day. According to these parameters, the system comes up with several possible options of how to defend against the incoming missile. These options may include using electronic warfare (EW), using counter-missiles, maneuvering, releasing decoys or even not doing anything at all.

To decide on the best option, the system runs a simulation of the incoming missile with each of the options. The output of the simulation is the probability that the missile will hit its target. If the simulation were deterministic, the system would just have to pick the option which generated the lowest probability that the missile will hit its target. However, since the simulation is of Monte Carlo type, using random variables for unknown parameters, the output is a random variable as well. Therefore, the system should run the simulation several times for each option and choose the option with the best mean.

If time were of no importance, the system would have to run the simulation many times for each option to decide which option is the best. Unfortunately, since the missile is on its way, the system can only run the simulation a limited number of times. We shall refer to the total number of simulation-runs the system can perform as the "budget" of the system.

Currently, the system divides its simulation budget evenly among the different options, then picks the option with the best mean. We shall refer to this as the "base case". While the base case may seem reasonable, there might be a better way to divide the budget among the options. For example, in a scenario where one of the options

generates results which are revealed to be clearly inferior, the system can eliminate this option early and allocate more time to decide among the remaining, better options.

B. THE PURPOSE OF THIS THESIS

This research attempts to find a better way of choosing the best option, using early elimination of inferior options. The purpose of this thesis is to provide system developers with the tools to incorporate such a process in the system, so that the decision would be more accurate. We introduce measures of performance and discuss the parameters which should be considered.

The problem is more important for systems operating in situations in which time is limited and the decision process is automatic. If the decision is not automatic and the operators have enough time to decide which options to eliminate, they can gain "intuition" of the situation and make calculations and comparisons accordingly.

C. SEQUENTIAL COMPARISON TESTING

The typical way to decide on the best option among several is to provide as many observations as possible for all the options, calculate their means and select the option that appears to be the best. In this paper we are looking at a method which compares the different options while the observations are being generated, which influences the selection process itself.

Testing and elimination can take place at different times during the process, for example:

- Two steps – generate 50 percent of the observations, test for elimination and generate the rest of the observations for the kept options.
- Four steps – generate 25 percent of the observations, test for elimination, generate another 25 percent of the observations and so on.
- Sequential – after generating at least one observation for each option, generate another observation for an option, test for elimination, generate another observation, test for elimination and so on.

In this thesis we consider sequential processes as a means of finding the best option. Unlike many such processes in the statistical literature, however, we impose a constraint on the number of observations that can be made. If the budget is exhausted and a single option has not been identified as being the best, we select the best of all options that are alive at that point, where "best" refers to the option with the smallest or largest sample mean depending on the context.

D. TRADE-OFFS IN EARLY ELIMINATION

In order to save as many simulation-runs as possible, we would like to eliminate inferior options as soon as possible. However, eliminating options too early might result in a false elimination of the best option. How early is early enough, but not too early? The answer to this question depends on:

- The budget. A bigger budget allows us to postpone eliminations until we are more confident about them.
- The number of options. With more options to choose from, we would like to eliminate options earlier, which might leave us with more good options towards the end.
- The results. The farther apart the sample means of the different options are, the more confident we are in making eliminations. Here "distance" is expressed in terms of the differences between option means and the variability of simulated observations.

E. DISTRIBUTION OF RESULTS

In order to understand the complexity of our problem, let's assume a simple scenario in which the results for each of the options are drawn from normal distributions with a known common variance. All we need to find out is which option has the best mean. In this thesis, the best option refers to the option with the smallest mean. We first apply a straightforward, intuitive method.

Since we know the variance, after a few observations we can estimate the mean of each option, and bound it within a confidence interval. As an elimination rule, we can

eliminate any option for which the confidence interval of its mean does not overlap with the confidence interval for the best option so far. Figure 1 illustrates how this intuitive elimination rule is applied:

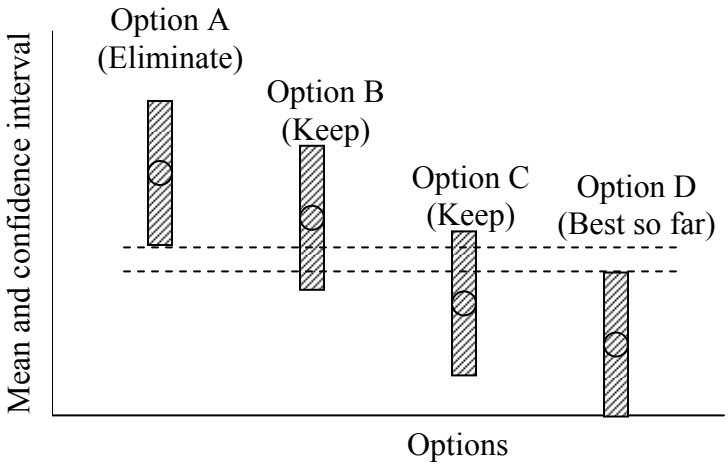

Figure 1. An intuitive elimination method based on confidence intervals for sample means.

But how large should the confidence intervals be? Should we use the typical 95 percent confidence level, or 90 percent or even 70 percent, or should we go as high as 99 percent? The answer depends on the budget and on the number of observations we have – with more budget and fewer observations we can postpone any elimination until we are very confident about it, hence using a wider confidence interval.

Now consider a problem in which we do not know the variances of the observations, or the shapes of their probability distributions. To handle this situation we will present several processes with different parameters, which the user can adjust for the system at hand. We assume that the simulation is accessible for the developer, so that it is possible to try these processes as much as needed, changing the parameters to best suit the problem.

F. OUR TOOLS

In order to test the different processes and to compare them with the base case, we wrote a computer program in Java to simulate data for different test scenarios, and to process the results using the methods we tested.

The program generates results according to normal and beta distributions, as well as from a simple unclassified simulation of the anti-missile system, which we developed according to information that is available to us, and using our own understanding of the problem. The program processes the results according to each method we want to test, as well as the base case. When a method declares which option is the best, the program compares it to the true best option. This process is repeated many times with different parameters for the methods, with different probability distributions, and under different budgets.

G. HEURISTIC APPROACH

An elimination process can be designed using a heuristic approach. We can build many different methods to choose the best option under different numbers of options, budget constraints and distributions of the options, and conduct a "contest" between the methods, comparing the results for each scenario with the base case. The best method would be chosen for each scenario.

The problem with this heuristic approach is that the number of parameters we need to change is very large, and if we want to try each parameter with several different values then we would have too many options to try. Therefore, we decided to use existing methods, even if they are not designed for this problem, which we then change and calibrate to suit our needs.

H. EXISTING METHODS

We focus on a class of techniques from the statistical literature known as "Sequential Multiple Comparisons with the Best". We shall refer to this as the SMCB process. A formal description of the SMCB process can be found in Hsu and Edwards (1983). We found several versions of the SMCB process, each fitting a different set of assumptions. A more detailed discussion of the SMCB process is deferred to Chapter II.

The SMCB process deals with a slightly different problem than the one we described: how to eliminate inferior options such that the probability of eliminating the best option by mistake is small. The process is not limited by any budget, and is not designed to be the fastest way to do this elimination. The SMCB process uses a

probabilistic method which sequentially gathers observations and uses this information to eliminate options, while the procedures make sure that the probability of eliminating the best option is controlled at a small value. The SMCB process considers it a success when the best option is not eliminated, and it does not force the selection of one of the remaining options when the budget is exhausted. In our research problem, however, the budget cannot be ignored, and only one option must be selected.

I. BUDGET

Suppose that our budget allows for only a limited number of simulation runs. There are two ways to look at this limitation:

1. The number of runs is limited to a certain number (e.g. 100 runs).
2. The faster the algorithm chooses the best scenario, the better. This way, each run should have a penalty or cost associated with it.

The first case can be considered as a special case of the second one, where the penalty for each run is a step function, with zero for each run below the limit, and infinity for each run over it. Since we do not know the trade-off between the number of runs and the quality of the decision, we will consider only the first case, where the number of runs is simply finite. In future work, one can look at a more complex cost function of the budget.

J. RESULTS

Compared to the base case, SMCB processes have an advantage in being able to eliminate obviously inferior options early. But this advantage quickly diminishes as the separation between the best option and inferior options decreases. Based on the results presented in Chapter III, we cannot make a conclusive statement about the advantage of using SMCB processes for moderately separated options. The sequential methods show potential under some combinations of test scenarios and budgets, but it seems that in order to unlock this potential the system developer needs to know more about the distributional properties of the simulation data than may be available.

Although our results show that the SMCB processes we consider are not always better than the base case, there may be other processes which perform better. We find that the SMCB processes do not eliminate options fast enough, and they do not take the budget into account when deciding whether or not to eliminate an option. On the other hand, as we noted above, the base case never eliminates any option, no matter how bad it is. However, extending the advantage of sequential methods beyond this case remains a challenging problem.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. EXISTING SOLUTIONS

We searched the statistical literature for existing solutions to our problem of choosing the best option with a fixed budget of observations. We did not find any such methods, but we did find methods for addressing a related problem: how to eliminate options while controlling the probability of eliminating the best option. These methods tend to be conservative, failing to eliminate inferior options quickly because there is no penalty for failing to do so.

We took these processes and modified them so that they fit our problem. In Chapter III we describe several other methods which we believe might provide better solutions.

B. THE SMCB PROCESS

This section continues the description of the SMCB process from Chapter I, showing its mathematical formulation as well as different methods to construct the thresholds.

The objective of the process is to eliminate options from a set of options, so that the probability of eliminating the best option is not greater than a specified probability P^*. We define the best option as the one with the smallest mean, for example, in the case of the anti-missile system we are looking for the option which generates the smallest probability of the missile to hit the ship.

The SMCB process operates in stages – we start by generating one observation for each option; this is the first stage. Each time we add another observation to each of the live (not-eliminated) options we advance one stage. This means that all the live options have n observations at stage n.

Before advancing to the next stage, the process tries to eliminate options. Eliminated options do not call for any more observations, and cannot return at a later stage. To eliminate an option, the process compares each option with the best option at

that point – if the difference is too big, that option is eliminated. The main issue in designing a SMCB algorithm is choosing the correct thresholds for elimination.

There are several methods of choosing these thresholds in the literature, depending on the distribution of observations: whether it is normal, and whether or not we know the standard deviation. Formulas for these thresholds are provided later on in this chapter.

1. General SMCB Formulation

Consider a scenario where we have k different options represented by (π_1, π_2… π_k), and we are interested in choosing the option with the smallest mean (with minor changes we can look for the largest mean or we can use a different statistic).

When we start the process all the options are relevant (none are eliminated). We shall refer to non-eliminated options as "live" options. We shall denote the set of live options at stage n as Π_n. Notice that since eliminated options cannot return at a later stage: $\Pi_n \supseteq \Pi_{n+1}$.

At the first stage we have one observation for each live option. Likewise, at stage n we have n observations for each live option. At each stage (we shall denote the stage number as n) we check which of the options should be eliminated before proceeding to the next stage:

a. We mark the option with the best sample mean so far. This option will not be eliminated. Assume that option j is that one.

b. For each of the remaining live options, calculate the difference between their sample mean and the sample mean of the best option:

$$\delta_i = \hat{\mu}_i - \hat{\mu}_j \qquad i \neq j, i \in \{i : \pi_i \in \Pi_n\}$$

c. Eliminate any option for which δ_i is greater than the threshold (C_n) at this stage.

d. If more than one live option is left, add another observation for each live option, and repeat the elimination process.

We can limit the SMCB process by a maximum number of stages or a maximum number of observations, choosing the option with the best mean among the remaining live options when the limit is reached. We made a slight modification to the process: if only two options are left alive and the budget is not yet fully used, the process will not eliminate another option before the budget is fully used.

The thresholds for elimination become smaller and smaller with each stage, making eliminations easier. Figure 2 illustrates how the thresholds decrease with the number of stages for an SMCB process based on normal distributions with known variances:

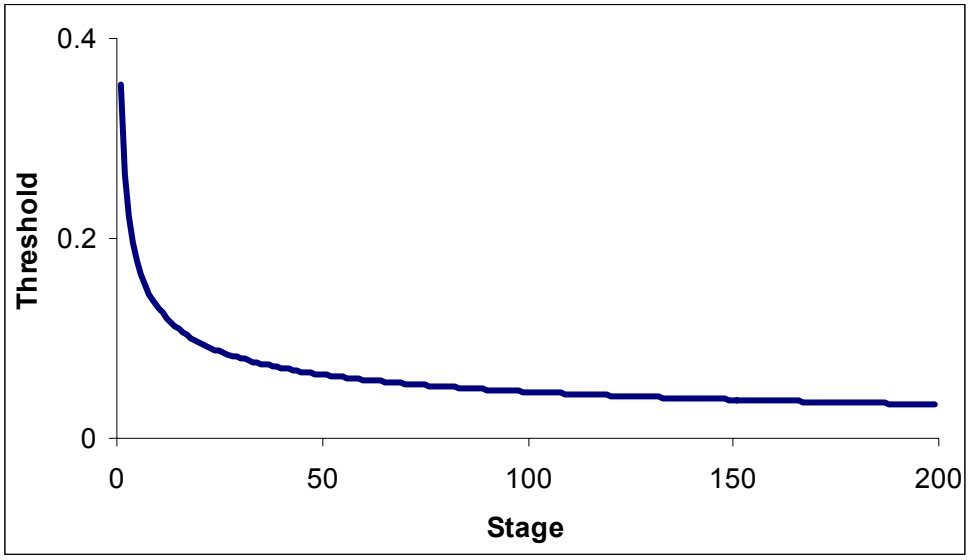

Figure 2. Threshold versus stage number for the SMCB process.
Based on five options with normal distributions and a known, common standard deviation of 0.1. Type I error is controlled at $P^* = 0.8$.

The following sections describe methods for selecting the elimination thresholds used by some of the SMCB processes in the statistical literature. These methods assume independence of observations in addition to the assumptions described below. Other methods for choosing the thresholds are described in Chapter V section A.

2. Thresholds for a Known Common Variance, Normal Distribution

This method, which is described in Swanepoel and Geertsema (1976), assumes that the observations for all options are normally distributed with a known common

standard deviation. The mathematical formulation of the problem is not intuitive, and is based on results from sequential statistical analysis.

Let k be the number of options, and P^* the probability of not eliminating the best option at any stage. We define $b = b(k, P^*)$ to satisfy the following equation:

$$1 - \Phi(b) + b\phi(b) + \frac{\phi^2(b)}{\Phi(b)} = \frac{1 - P^*}{k - 1},$$

where Φ and φ are the CDF and density of the standard normal distribution, respectively. Let

$$g = g(n, b) = \left(\frac{b^2 + \log n}{n} \right)^{1/2}.$$

The threshold at stage n is $\sqrt{2}\sigma g(n, b)$, where σ^2 is the common variance of all the options. We denote this method as $\text{SMCB}_{\text{known}}$, to recognize that this process assumes a known common variance.

3. Thresholds for an Unknown Variance, Normal Distribution

This method, which is also described in Swanepoel and Geertsema (1976), assumes that the observations for all the options are normally distributed with unknown and possibly different standard deviations, which must be estimated in the process.

Let k be the number of options, and P^* the probability of not eliminating the best option at any stage. We define $a = a(k, P^*)$ to satisfy the following equation:

$$0.5 - \frac{\arctan(a)}{\pi} + \frac{a}{(1 + a^2)\pi} = \frac{1 - P^*}{2(k - 1)}$$

Let $t = (1 + a^2)^2 / 2$, and

$$h(t, n) = [(tn)^{\frac{1}{n}} - 1]^{1/2}.$$

The quantity $h(t,n)$ is one component of the threshold, which is multiplied by another component which takes the observations' estimated variances into account. This component is calculated using the differences of each pair of options. Denote two of

these options by i and j. At stage number n, $\overline{X}_i(n)$ is the average over all observation for option i. Letting $X_{i\beta}$ denote the β^{th} observation under option i, we define

$$H(j,i,n) = \left[\frac{1}{n} \sum_{\beta=1}^{n} \{(X_{j\beta} - X_{i\beta}) - (\overline{X}_j(n) - \overline{X}_i(n))\}^2 \right]^{1/2}.$$

Finally, the threshold at stage n when comparing option i and option j is

$$C(n) = h(t,n)H(j,i,n).$$

Since we do not know the variances, it might happen that the best option has a large variance compared to the others, and this will make eliminations harder. Therefore, in this scenario we should compare all the different pairs, since eliminations might occur between two options which are not the best so far, even if none of them was eliminated by the best option so far. Figure 3 illustrates this – the bars on the left are the confidence intervals for the sample means of options A, B and C. The bars on the right are the threshold limits for each comparison between two options. Option A is eliminated by option B rather than option C, even though option C has the best (smallest) sample mean.

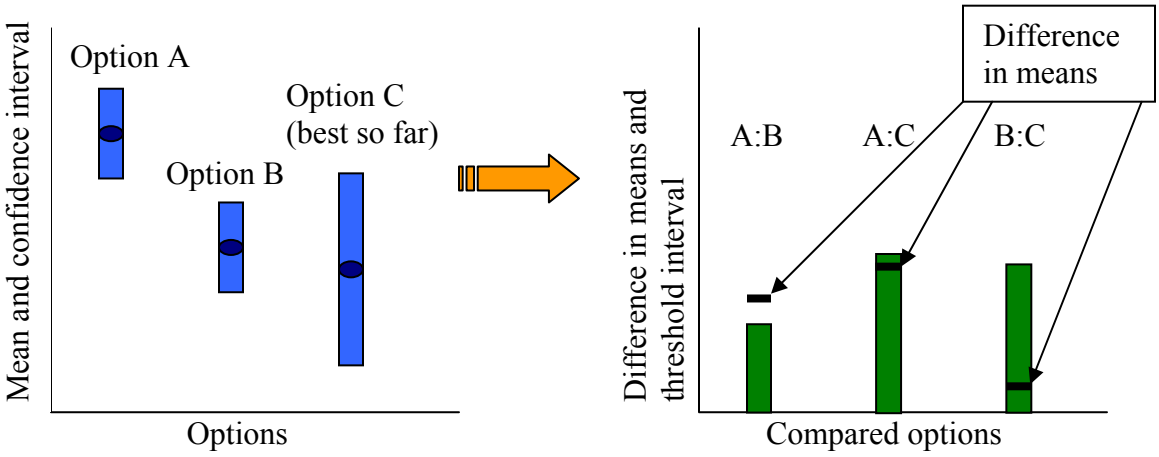

Figure 3. An illustration of the use of confidence intervals for selecting the option with the smallest mean.
Assuming normality and unknown, possibly different standard deviations.

We denote this method as $SMCB_{unknown}$, to recognize that this process assumes an unknown variance.

C. COMPUTATIONAL COMPLEXITY

The problem we are dealing with is one in which time is limited. Therefore, we should be concerned about the running time of the different algorithms – if any of the algorithms is too slow it might leave less time to generate observations. The running time of the algorithms depends on the total number of observations across all the options and the number of options. We shall denote the total number of observations as b (for budget), and the number of options as k. We shall follow the convention and present the running time of the algorithms by the dependence on the order of the parameters, O(parameter). In short, this is a common form to specify the running time of an algorithm according to the parameter which influences it the most. For example, if the running time is $30n^2$, then we say that it is of the order of n^2, or $O(n^2)$.

In the base case we need to calculate the sample mean for every option and find the option with the best sample mean. Calculating the sample mean for every option takes time in the order of b, and the time to choose the option with the best mean is in the order of k. Since the number of options is not greater than the budget, $k \leq b$, the total running time is $O(b)$.

With the sequential process we have a variable number of calculations at each stage. The number of stages depends on the budget and the number of options at each stage, and can be marked as b/k. At each stage we need to calculate the mean of each option. Although calculating the mean of n numbers takes $O(n)$, we can use the value from the previous stage to calculate the mean in a time of $O(1)$ at each stage.

Choosing the best option at each stage takes $O(k)$, and comparing this option to each of the other options takes another $O(k)$. So, if the number of stages is $O(b/k)$ then the total running time is also $O(b)$.

In case we need to compare each pair of live options at each stage, the running time for that is $O(k^2)$, and the total running time would be $O(kb)$.

To summarize, we can say that the running time for the base case is $O(b)$ and for the sequential procedures it is either $O(b)$ or $O(kb)$. However, calculating the order of the running time might be deceiving, as the actual running time of the sequential procedure

may be several times that of the base case, even with $O(b)$. To say whether this difference is significant or not requires a comparison between the running time of the simulation and that of the sequential process.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

A. THE SIMULATION COMPUTER PROGRAM

In this section we describe the computer program we developed to test different algorithms for selecting the best of a collection of options. Our description does not include features and attributes which are not important to the understanding of the program. The computer program is available from the author and advisor of this thesis.

1. Overview

The program generates observations according to the test scenarios described later in this chapter. Each time the program generates observations for a single test scenario, it runs the base case and sequential algorithms on these observations. Each algorithm chooses the best option according to its rules, and a comparison is made with the true best option. This process is repeated many times for each test scenario. The test scenarios can come from the normal distribution, the beta distribution or from a simplified simulation of missile trajectories that we describe below.

2. The Missile Trajectory Simulation

This simulation assumes that there is a missile approaching a ship. The ship can put decoys around it, hoping that the missile will choose one of the decoys instead of the ship. Each of the decoys, as well as the ship, has a signal strength (represents the radar cross section, for example), and the missile randomly chooses one of these elements and homes in on it.

The probability of choosing each element is proportional to its signal strength times the cosines of its angle from the missile's axis. The missile's flight direction is randomly chosen around the element to which it is homing. The probability that the missile will kill the ship is estimated from the simulation, using the minimum distance of the missile from the ship. Figure 4 illustrates this:

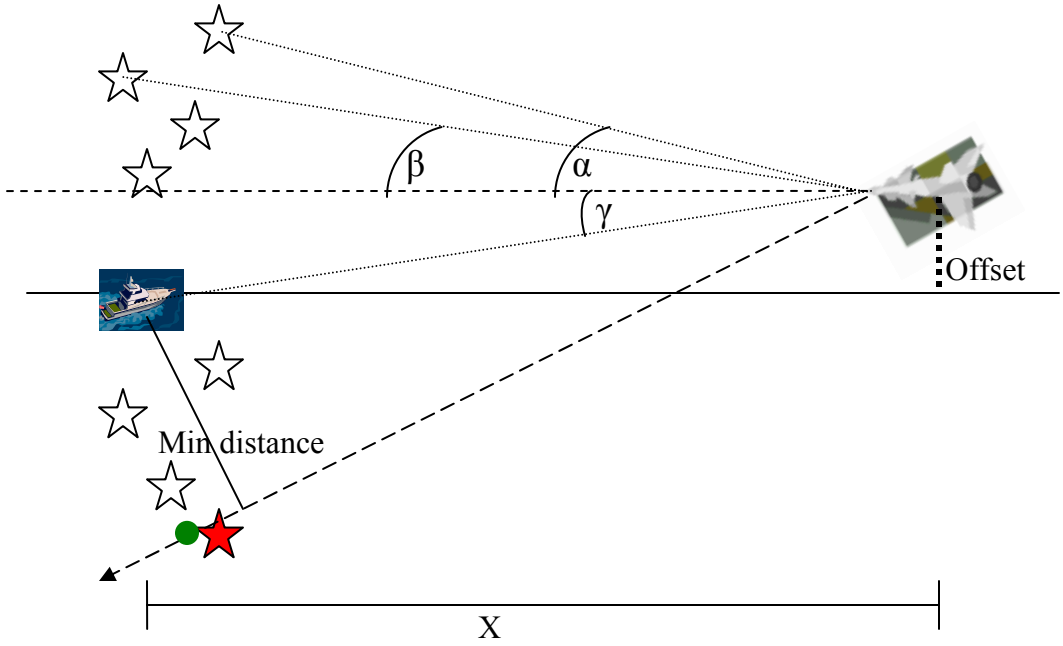

Figure 4. An illustration of the missile trajectory simulation.
A missile starts at a distance X from the ship at a random offset. The missile chooses a decoy according to the element's signal power and angle from the missile's axis. The chosen decoy is marked by the red star, and the missile actually heads to the green point, randomly chosen next to it. The probability of kill is calculated according to the minimum distance of the missile from the ship during its flight.

The advantage of using this simulation is that it might generate observations according to a distribution that is very different from the normal or beta distributions. On the other hand, it is more difficult for us to predict how well a method would perform in a given scenario. One must also have the simulation in order to reproduce the results, whereas it is much easier in the case of the conventional distributions.

3. Determining Which is the Best Option

In a simulation using a known probability distribution, we know in advance which is the best option, and we can compare the results of the algorithms with this option. In other types of simulation this may not be possible. To determine which is the best option, the program generates ten thousands observations for each test scenario (much more than the budget), calculates the mean for each option and chooses the best one. Actually, we use this method for the normal and beta distributions as well.

B. MEASURES OF PERFORMANCE

Our simulation generates observations for all the options, each algorithm chooses the best option according to its rules, and a comparison is made with the best option. In this section we describe several measures of performance that can be used to compare algorithms for selecting the best option:

1. The Fraction of Times an Algorithm is Correct

This tells us the fraction of times the algorithm chose the best option, but it does not show us how far from the best the algorithm was when it chose the wrong option – it might have chosen the second best, but it also might have chosen the worst one. We shall denote this measure of performance as MOP1.

2. The Average Error

Each time an algorithm chooses an option we calculate the difference in means between this option and the best option (i.e. if it chooses the best option, the difference is zero). The average of these differences is our second objective function. We shall denote this measure of performance as MOP2.

3. The Average Number of Observations Needed

The average number of observations needed to get the correct option a certain fraction of the time. Although this is an important measure of performance for time-critical applications, we did not use this measure of performance in our research.

C. SMCB VERSUS THE BASE CASE

The SMCB process might not be any better than the base case – consider a case where the thresholds for elimination are very large, such that no options are eliminated until the entire observation budget is used. In that case, the budget is equally divided among the options and the selection is made for the option with the best mean. This case is the same as the base case, where the observations budget is divided equally among the options at the beginning, and therefore the result would be the same. However, the SMCB process would have imposed greater computational costs than the base case, with no additional benefit.

Likewise, the SMCB process might be better or worse than the base case, depending on several parameters, some of which are described in Chapter I and some which are described later in this chapter. One of these parameters is the existence of very bad options which can be eliminated early. In order to make it clearer, we consider three types of scenarios:

- All the options which are non-optimal are statistically the same. If that is the case, either the non-optimal options are close to the optimal one and no options will get eliminated early or they are so far from it that using the base case is good enough. In either case, the results are likely to be close to those of the base case; hence we label this the "worst case" scenario.
- There is a combination of non-optimal options, some of which are close to the optimal and others far from the optimal. The options are likely to be eliminated one by one, gradually leaving only the options which are close to the optimal one. We label this the "gradual" scenario.
- Half of the non-optimal options are close to the optimal one and unlikely to get eliminated early, while the other half are far from it and might get eliminated early. We label this the "half-and-half" scenario.

The next chapter describes the test scenarios we chose to use for the evaluation of the performance of the SMCB processes compared to that of the base case.

D. TEST SCENARIOS

In order to test the algorithms and to compare their efficiency to that of the base case, we need to test it on data from the actual system simulation. Unfortunately, we do not have access to the actual simulation, so we developed a substitute.

We used three types of sources for randomly generated data: normal distributions, beta distributions and a simplified interpretation of the actual simulation, which we built according to our understanding of the problem. For each source we decided on several test scenarios, each with a different number of options to choose from, and different

parameters for those options. We chose the test scenarios so that they will span the space of options with the following dimensions:

1. Number of Options and the Budget

We expect test scenarios with more options and a smaller budget to show a bigger difference between the base case and the sequential algorithms.

2. Separation of Options

We define the separation between two options as the difference in means divided by the standard deviation:

$$\Delta_{1,2} = \frac{\mu_1 - \mu_2}{\sigma}.$$

If the options have different standard deviations, consider:

$$\Delta_{1,2} = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}\sqrt{2}\,.$$

This measure is directly related to the probability that for a single observation from each option, observation from the better option will be greater than that of the other one. We shall refer to this as the separation between the options. We considered two factors regarding the separation:

a. The separation between the best option and the second best.

b. The distribution of the means of good and inferior options.

Other than these guiding rules, the construction of the test scenarios is arbitrary.

We divide the test scenarios in three groups according to the type of source: normal distribution, beta distribution and the simplified anti-missile simulation. The next three sections discuss the test scenarios for these sources. In addition we use several other test scenarios, which are described in Chapter III.

E. TEST SCENARIOS FOR THE NORMAL DISTRIBUTION

For the normal distribution we use the following parameters in order to build the test scenarios:

1. Number of Options

We use both k=5 and k=9 options in our simulations. While these numbers are arbitrary, we think that having five options is the minimal scenario which is interesting for this problem and a scenario with nine options represents one with many options.

2. Separation

We use separations of $\Delta_{1,2}$= 0.1 , 0.2, 0.3 … up to 1.0. This is an appropriate range for the budget we use and the precision we require – a smaller separation requires a larger budget, while a larger separation results in a very small probability of mistake in choosing the best.

3. Distribution of the Separation of Options

We consider three simple cases:

a. One best option, all the other options are the same, separated from the best by the specified separation. This corresponds to the "worst case" scenario described in section C, above.

b. The best is separated from the second best by the specified separation, from the third best by twice the specified separation, from the fourth by three times the specified separation and so on. This corresponds to the "gradual" scenario described in section C.

c. Half of the non-optimal options are separated from the best by the specified separation, while the other half is separated by four times the specified separation. This corresponds to the "half-and-half" scenario described in section C.

4. Standard Deviation

All but one of the scenarios we consider have a common standard deviation for all the options.

The total number of test scenarios is:

$$2(options)\times 10(separations)\times 3(distributions\_of\_separations)=60.$$

The extra test scenario with the varying standard deviation is a variation of the "half-and-half" case, but for all the options the standard deviation is changed so that the ratio μ/σ is kept constant.

5. Budget

Our problem of selecting the best option is limited in running time, which limits the number of observations we can generate. We define the total number of observations available as the budget. Since in the base case the total budget is divided evenly among the options, we decided to set the budget levels according to the number of options in a test scenario.

We consider four levels of budget per option: 5, 10, 20, and 40, which translate to total budget over all the observations of 25, 50, 100 and 200 for the scenarios with five options, and 45, 90, 180 and 360 for the scenarios with nine options.

F. TEST SCENARIOS FOR THE BETA DISTRIBUTION

The beta distribution is characterized by two parameters: α and β. The distribution is concentrated in the [0,1] interval, with a mean of $\mu=\alpha/(\alpha+\beta)$ and a variance of

$$\sigma^2=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

The shape of a beta distribution is defined by the parameters α and β, and is asymmetrical for $\mu \neq 0.5$. Due to these properties, the beta distribution is commonly used as an approximation for unknown probability distributions.

We decided to use the same scenarios as those for the normal distribution, with the following necessary changes:

1. The Mean

The beta distribution is concentrated on the interval [0,1] and so its mean is between 0 and 1. Changing the location of the mean in this interval also changes the

shape of the distribution. We use two scenarios, one with the means around 0.5, signifying a symmetric bell-shaped distribution, and one with the means around 0.1, for an asymmetric distribution.

2. The Standard Deviation

In order to keep the same separation values as we did in the normal case, we use a standard deviation of 0.1, instead of 1.

3. The Parameters

The beta distribution is specified by two parameters: α and β. We translated the means and standard deviation into these parameters according to these equations:

$$\beta = \frac{\alpha}{\mu} - \alpha \text{, and } \alpha = \frac{(1-\mu)\mu^2 - \sigma^2\mu}{\sigma^2}$$

We test these scenarios with different budgets, the same way as for the normal distribution scenarios.

G. TEST SCENARIOS FOR THE MISSILE TRAJECTORY SIMULATION

The test scenarios for the simulation are more arbitrary than the previous ones. Since we cannot define the mean or the standard deviation of the results a-priori, we need to change the parameters in an arbitrary way.

Description of the missile trajectory simulation is given in this chapter in section A.2. The parameters we can change are the number of decoys, the signal strength of each decoy and that of the ship, the location of the decoys, the distribution of the offset and the missile's accuracy. Each option represents a set of values for these parameters, and each scenario has several options, one of which is the best. These are the scenarios we use:

- In the first scenario, we use a different number of decoys for each option, cutting the uppermost decoys first.
- In the second, third and forth scenarios we change the signal strength of the decoys compared to that of the ship, making their signal strength weaker and

weaker for every option. In the second scenario this difference is small, and it gets larger in the third and forth scenario, hence it is more difficult to discern the best option in the second test scenario and easiest in the forth.

- In the fifth scenario we change the accuracy of the missile.
- In the sixth scenario we change the location of the decoys, placing them closer and closer to the ship.
- In the other four scenarios we mix options arbitrarily from the six scenarios described above.

For the $SMCB_{known}$ method we need to specify a standard deviation value. We use σ=0.5 since it is an upper limit for the real value.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS

A. PREFACE

The purpose of this chapter is to show whether the SMCB methods are better than the base case. The results are not conclusive about this, and the advantage depends on several parameters such as the number of options and the budget.

The next section describes the expected results from the SMCB methods as well as from the base case. Samples from the raw results follow that section, and then a discussion about the actual advantage of using the SMCB methods.

B. EXPECTED RESULTS

In this section we compare the different selection algorithms, and we describe the qualitative results we expect to see for any of the selection processes. The discussion in this section is qualitative in nature and is intended to help the reader understand results presented later in this chapter.

1. Performance versus Separation

We use two measures of performances:

a. Fraction of correct decisions versus separation (MOP1)

Reasonable algorithms should produce monotone, non-decreasing graphs – the greater the separation (see section III.D.2) the more often we are correct. The graphs should appear as shown in Figure 5:

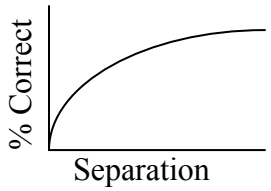

Figure 5. Depiction of the expected percentage of correct decisions (MOP1) versus separation for a typical selection algorithm.

b. Average difference error versus separation

The difference in means between the best option and the option chosen by the selection process can be used as a measure of distance. Because the selection process operates on random data, this distance is also random. For our measure of performance we take the average difference over the simulation. If the separation is very small, choosing a wrong option would result in a small difference from the best option, because the options are close to one another. If the separation is very large, the process would be correct most of the time, and the average difference would be very small. Therefore, a plot of the average difference versus separation should appear as shown in Figure 6:

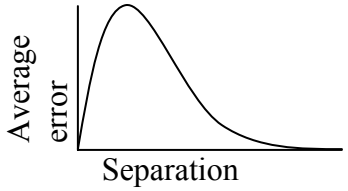

Figure 6. Depiction of the average error (MOP2) versus separation for a typical selection algorithm.

2. Performance versus Budget

When the budget is greater, we expect to get a better decision, resulting in a higher probability of choosing the best option, and a smaller average error. For the base case, when the budget is very large the probability of making the correct decision is close to one. When the budget is small, we expect the sequential processes to have better results than the base case. However, for the sequential processes we use, a false elimination can occur at some stage, and a large budget will not compensate for that. Therefore, for the sequential processes the limit is less than one. Figure 7 illustrates this:

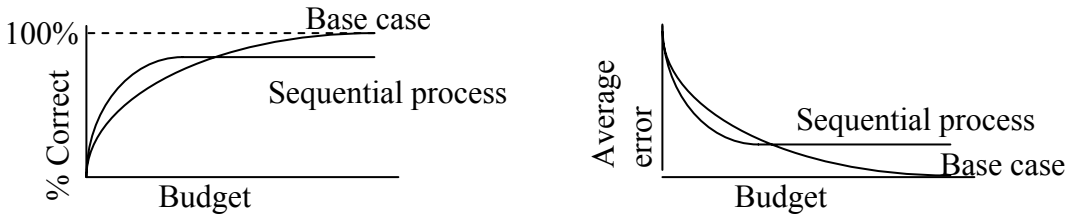

Figure 7. Expected performance versus budget, sequential and base case processes.

Figure 7 shows a scenario where the sequential process is better than the base case for a small budget, and worse for a large one. While this scenario is possible, it is also possible that the sequential process would always be worse than the base case, if it eliminates options too fast.

3. Performance versus Number of Options

If we fix the number of observations per option and add more options the base case will only do worse – there are more opportunities to choose a wrong option. While the sequential processes also have more opportunities to choose a wrong option, they also have more opportunities to eliminate bad options and use their saved observations for the other options. An extreme example would be to add an option which is so bad that it will get eliminated on the first or second stage, with practically no chance of being selected as the best one. In this example the base case is not affected, but the sequential processes will eliminate this option early and have a bigger budget for the other options; it is like changing the overall budget of the sequential process while leaving the base case budget fixed.

4. Performance versus Distribution of Observations

The sequential (SMCB) processes that we consider are based on an assumption that the data are normally distributed. However, these processes might be sensitive to changes in the type of distribution, especially distributions with asymmetric shapes. The base case should be more robust; i.e., less sensitive to these changes.

5. The Quality of the Processes

We check two types of sequential processes: one in which the standard deviations of the observations are known and one which they are not. We expect the first to be better, since it is optimized for the case of a known standard deviation. It is difficult to say which of the different P^*'s performs better. However, we expect that as P^* becomes larger, it would eliminate less options, and therefore be closer to the base case, which does not eliminate any options until the entire budget is expended.

6. The Combination of Inferior and Superior Options

Section III.E.3 describes the three ways we distribute the inferior and superior options of a test scenario: "worst" case, "gradual" case and "half-and-half" case. We expect the sequential methods to do better than the base case in scenarios where there are extremely inferior options that can be eliminated early. In the "gradual" and "half-and-half" cases, there are such extremely inferior options and therefore we expect to see the advantage of a sequential method under these cases. In the "worst" case, on the other hand, all inferior options have the same mean, so if the separation between the best and these inferior options is too small they will not be able to discard options before the budget is exhausted. Thus, the sequential methods will work just like the base case, and if the separation is big enough for the sequential methods to eliminate an option, then it is probably also easy for the base case to distinguish the best option. Taking into account the fact that the sequential methods might falsely eliminate the best option, the results of the sequential methods can either be slightly better or worse than those of the base case.

C. SIMULATION RESULTS

Each test scenario generates a comparison between the base case process and the SMCB processes. We use two SMCB processes, one which assumes a known standard deviation, $SMCB_{known}$, and one which assumes unknown standard deviation, $SMCB_{unknown}$, and four values of P^* for each of these two processes. This results in hundreds of comparisons. For obvious reasons we only show a summary of the results in this paper, and a few examples of the results. The examples are taken from three test scenarios using normal distribution. The next section discusses the overall of the results.

1. The Normal Distribution, Five Options, Budget of 100 Observations, "Gradual" Separation

The following graph shows the portion of time the different processes chose the best option for this test scenario, for different values of separation:

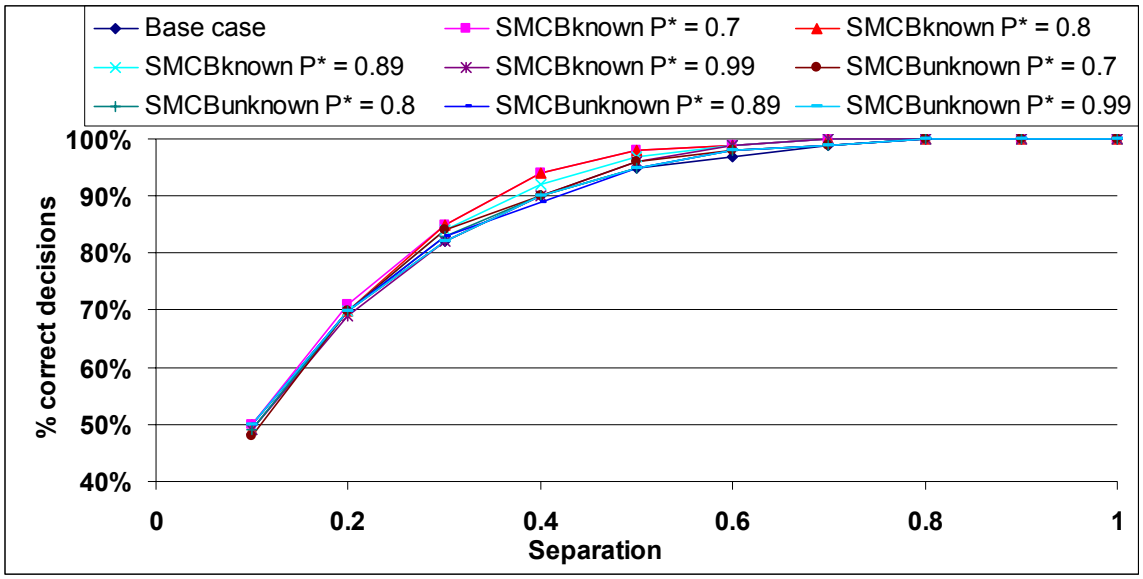

Figure 8. Percentage of correct decisions (MOP1) versus separation. Different methods with a budget of 100 observations, observations from the normal distribution, "gradual" separation case.

There is a very small difference, if any, between the different processes. All the processes are doing better with a bigger separation, as expected. It is difficult to see anything else from this graph, and a better presentation is discussed later, in section 3 of this chapter.

2. The Normal Distribution, Five Options, Budget of 100 Observations, "Half-and-Half" Separation

The following graph shows the average error (see MOP2 in III.B.2) of the different processes under this test scenario; we reduced the number of processes for the clarity of presentation.

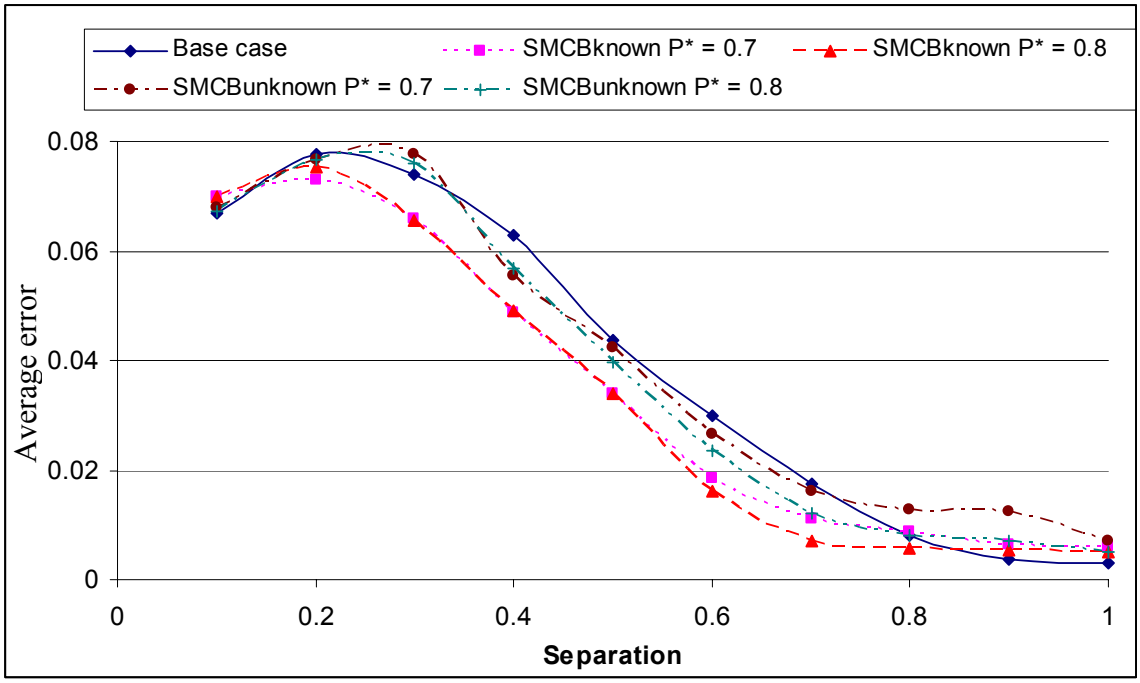

Figure 9. Average error (MOP2) of different processes versus separation. Different methods with a budget of 100 observations, observations from the normal distribution, five options, "half-and-half" separation case.

Each of the processes has a maximum that occurs somewhere between the smallest and largest separation, as expected (see B.1.b). At the beginning, the base case is a bit better than the other processes, and then it turns to be the worst, and returns to be one of the best when the separation is big enough.

3. Difference from the Base Case

It is not easy to see the details in the figures presented in section 2. To make the details clearer we present the measures of performance for the sequential (SMCB) methods as differences from the base case:

- $\Delta_{MOP1} = MOP1_{sequential} - MOP1_{base\_case}$, and
- $\Delta_{MOP2} = MOP2_{sequential} - MOP2_{base\_case}$

Alternatively, one may argue that these differences should be expressed relative to the base case:

$$\frac{MOP1_{sequential} - MOP1_{base\_case}}{MOP1_{base\_case}} \text{ and}$$

$$\frac{MOP2_{sequential} - MOP2_{base\_case}}{MOP2_{base\_case}}$$

but then we have to deal with a possible division by zero. Note that the sequential (SMCB) methods perform better then the base case according to whether Δ_{MOP1} is positive and/or Δ_{MOP2} is negative.

After making these transformations, Figure 8 and Figure 9 are re-expressed as Figure 10 and Figure 11 below; we also reduced the number of processes for the clarity of presentation.

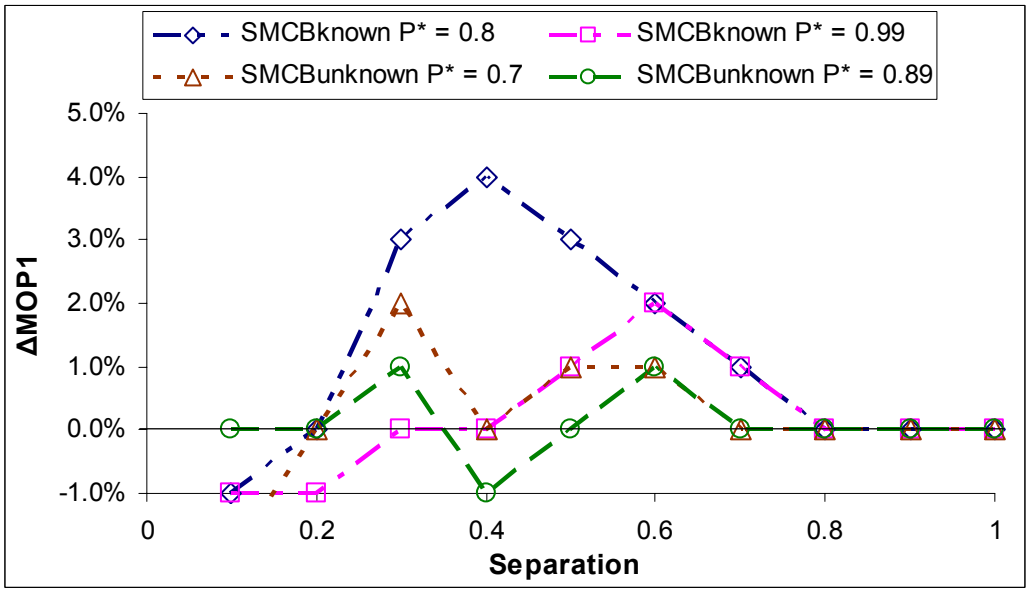

Figure 10. Percentage of time correct, difference from the base case (Δ_{MOP1}). Positive values are favorable for the SMCB methods. Budget of 100 observations, observations from the normal distribution, "gradual" separation case, five options. Data are the same as for Table 1.

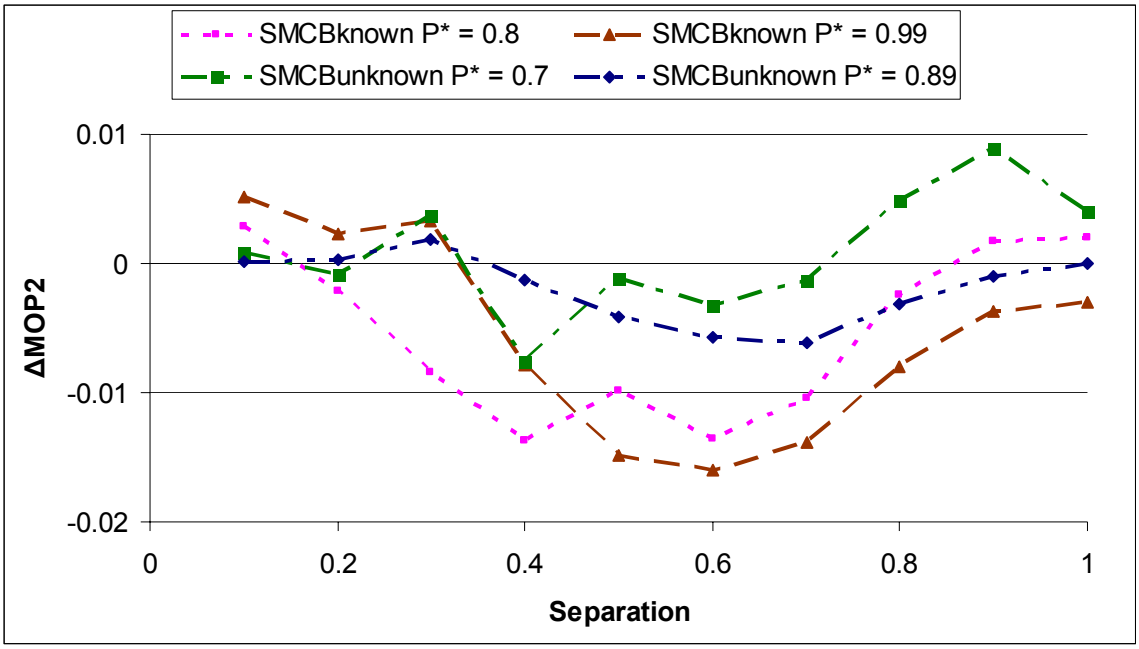

Figure 11. Average difference error, difference from the base case (Δ_{MOP2}). Negative values are favorable for the SMCB methods. Budget of 100 observations, observations from the normal distribution, five options, "half-and-half" separation case. Data are the same as for Table 2.

Figure 10 and Figure 11 can also be represented in a table form, where each row represents a value of separation in a test scenario and each column represents a sequential method with a specific *P** value. Table 1 and Table 2 show these figures in table form. The colors help to distinguish visually the cases where a method is worst than the base case (orange), slightly better than the base case (light green) or a lot better (dark green).

Separation	$SMCB_{known}$		$SMCB_{unknown}$	
	P* = 0.8	P* = 0.99	P* = 0.8	P* = 0.99
0.1	-1.0%	-1.0%	-1.0%	0.0%
0.2	0.0%	-1.0%	0.0%	0.0%
0.3	3.0%	0.0%	1.0%	0.0%
0.4	4.0%	0.0%	0.0%	0.0%
0.5	3.0%	1.0%	0.0%	0.0%
0.6	2.0%	2.0%	1.0%	1.0%
0.7	1.0%	1.0%	0.0%	0.0%
0.8	0.0%	0.0%	0.0%	0.0%
0.9	0.0%	0.0%	0.0%	0.0%
1.0	0.0%	0.0%	0.0%	0.0%

Table 1. Percentage of time correct, difference from the base case (Δ_{MOP1}). Positive values are favorable for the SMCB methods. Budget of 100 observations, observations from the normal distribution, "gradual" separation case, five options. Data are the same as for Figure 10.

Separation	$SMCB_{known}$		$SMCB_{unknown}$	
	P* = 0.8	P* = 0.99	P* = 0.8	P* = 0.99
0.1	0.0029	0.0052	0.0001	-0.0002
0.2	-0.0022	0.0023	-0.0007	0
0.3	-0.0085	0.0033	0.0018	0.0003
0.4	-0.0137	-0.0079	-0.0062	-0.0008
0.5	-0.0099	-0.0149	-0.0042	-0.0025
0.6	-0.0136	-0.016	-0.0063	-0.0036
0.7	-0.0104	-0.0139	-0.0054	-0.0034
0.8	-0.0024	-0.008	0	-0.0039
0.9	0.0017	-0.0037	0.0035	-0.0019
1.0	0.002	-0.003	0.002	-0.001

Table 2. Average difference error, difference from the base case (Δ_{MOP2}). Negative values are favorable for the SMCB methods. Budget of 100 observations, observations from the normal distribution, five options, "half-and-half" separation case. Data are the same as for Figure 11.

4. The Normal Distribution, Five Options, Different Budgets, "Worst Case" Separation

Figure 12 shows the percentage of correct decisions (MOP1) for the base case, $SMCB_{known}$, and $SMCB_{unknown}$ selection processes with P* = 0.8 and four different budget levels (25, 50, 100, and 200).

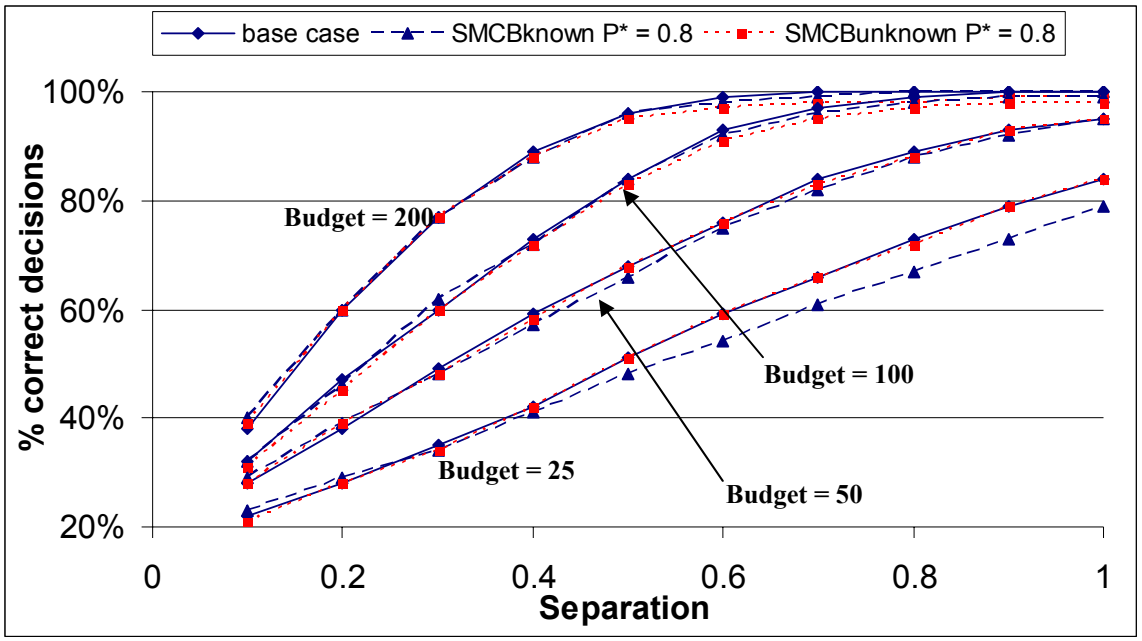

Figure 12. Percentage of correct decisions (MOP1) of different methods versus separation.
Budget levels of 25, 50, 100 and 200 observations, observations from normal distribution, five options, "worst case" separation.

Each of the processes shown in Figure 12 performs better with a larger budget. We can see that in this test scenario the base case is a bit better than the SMCB process when the budget is small, and this advantage gets smaller as the budget gets larger.

D. THE ADEQUACY OF SEQUENTIAL METHODS

The simulation results reveal cases in which the sequential methods do better than the base case, and others in which they do worse. Therefore, we can say that there is potential for using the sequential methods, but assigning the correct method might not be easy. This section describes the strengths and weaknesses of using the sequential methods, as seen in the results. Since the test scenarios we use do not represent the actual

problem, we put the emphasis on the points which we think are important for future use of these methods, rather than on the numerical results.

In this section we describe the performance of the SMCB methods compared to that of the base case under the same conditions (same test scenario), not their absolute performance. It is possible that other sequential methods behave differently than the methods we introduced, and a complete analysis is needed for any other method.

1. Test Scenarios

The sequential methods' advantage is strong in some of the test scenarios, while in others it is weak to the point of turning into a disadvantage.

The advantage can be seen in test scenarios containing options which are extremely far from the best option. The SMCB methods perform the worst in the "worst case" test scenarios, as expected. The performances in the "gradual" and "half-and-half" are usually better than the base case and very close to one another.

2. Budget

We expected to see the advantage of the sequential methods relative to the base case in scenarios with a small observation budget, and a disadvantage for scenarios with a large budget. However, the results show that the sequential methods may start with a disadvantage for small budgets, an advantage for bigger budgets and a disadvantage again for even bigger budgets. Figure 13 shows an example for the relative error versus the budget:

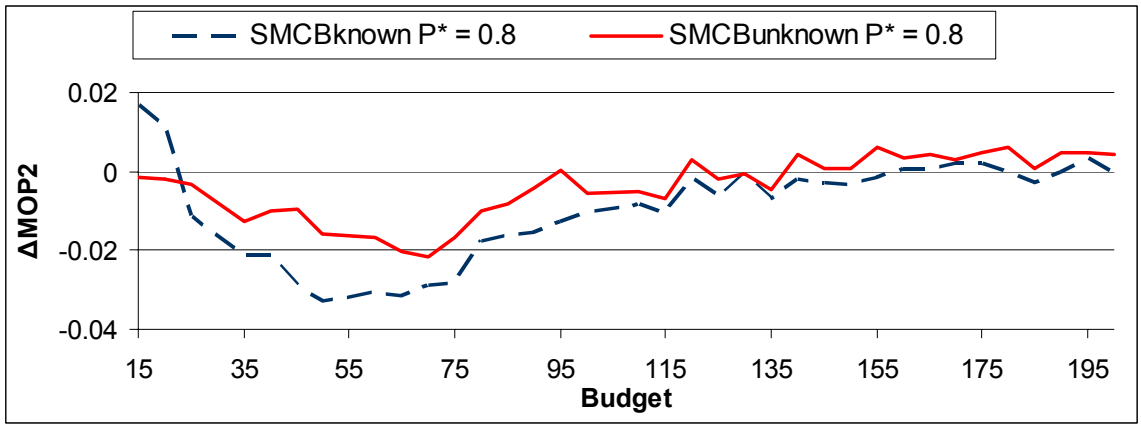

Figure 13. The relative error versus the budget (Δ_{MOP2}), separation of 0.7. The errors of the methods after subtracting out the base case errors. Negative values are favorable for the SMCB methods. Observations from a normal test scenario, five options, "half-and-half" separation.

The points at which the disadvantage turns into an advantage and vice versa depend on the test scenario, and on the separation for this test scenario. Figure 14, Figure 15 and Figure 16 illustrate this dependence.

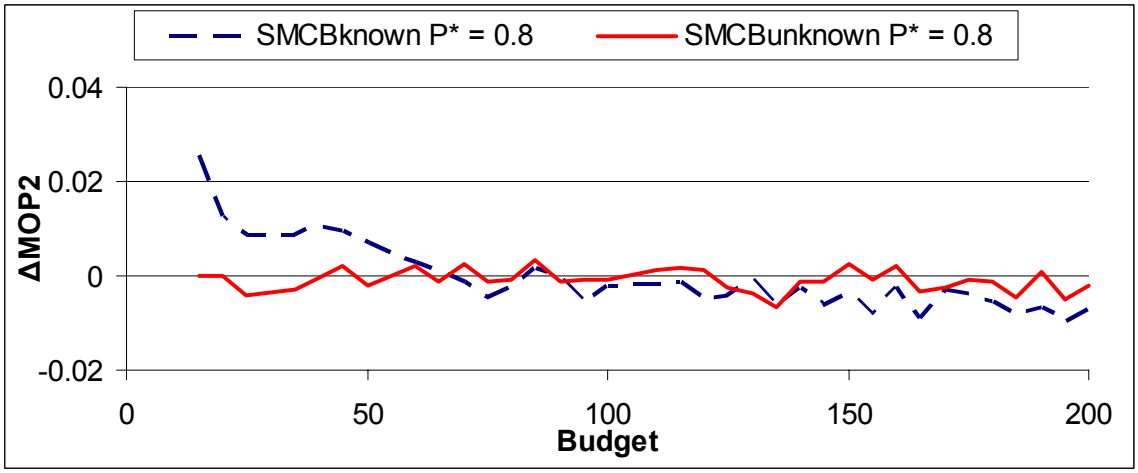

Figure 14. The relative error versus the budget (Δ_{MOP2}), separation of 0.2. The errors of the methods after subtracting out the base case errors. Negative values are favorable for the SMCB methods. Observations from a normal test scenario, five options, "half-and-half" separation.

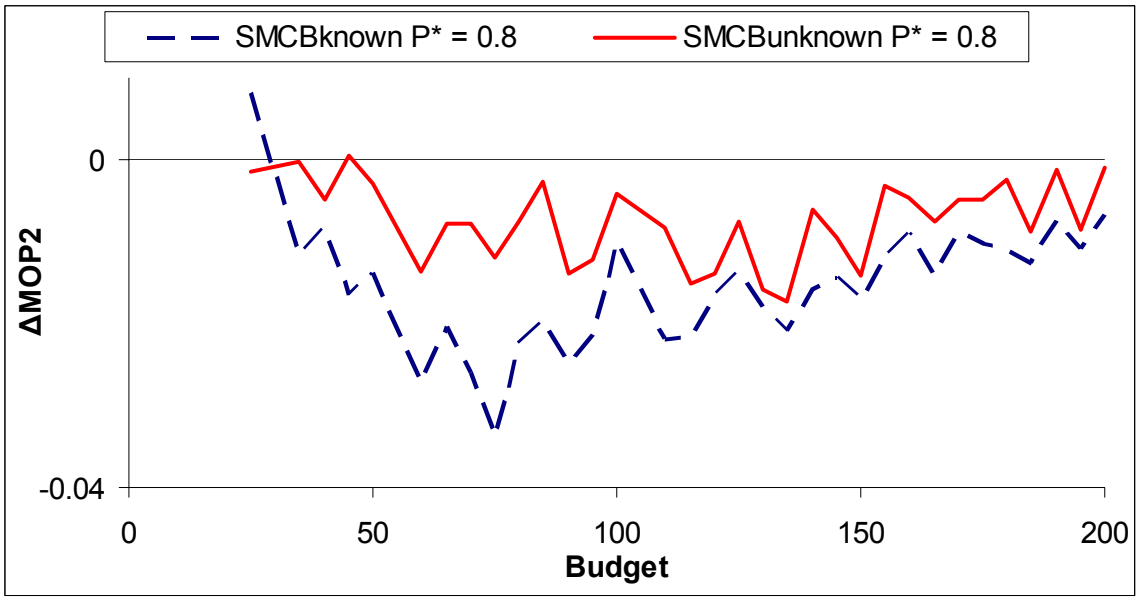

Figure 15. The relative error versus the budget (Δ_{MOP2}), separation of 0.5. The errors of the methods after subtracting out the base case errors. Negative values are favorable for the SMCB methods. Observations from a normal test scenario, five options, "half-and-half" separation.

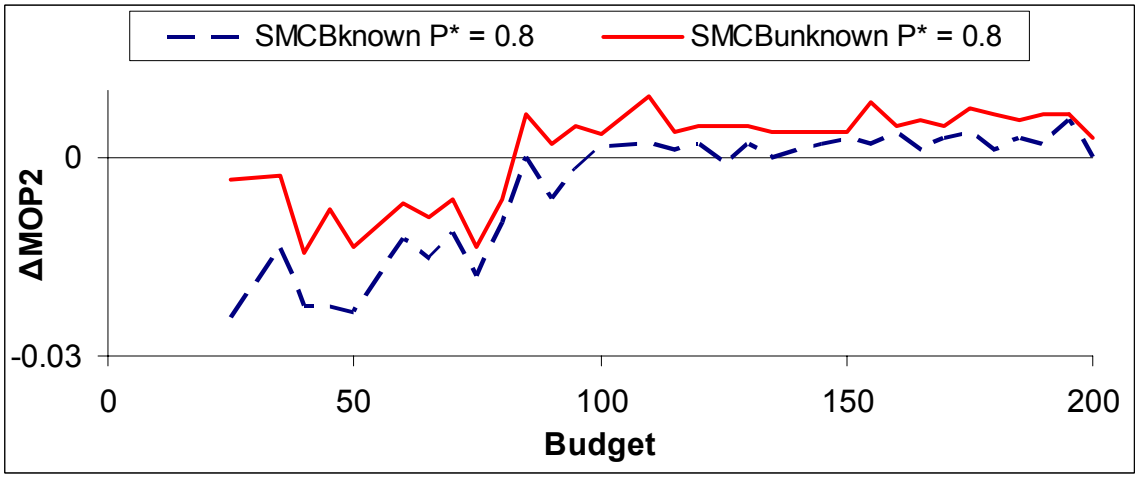

Figure 16. The relative error versus the budget (Δ_{MOP2}), separation of 0.9. The errors of the methods after subtracting out the base case errors. Negative values are favorable for the SMCB methods. Observations from a normal test scenario, five options, "half-and-half" separation.

In the eyes of a system developer, the results of our research suggest that the existence of very bad options and a small budget are not the only criteria for the

usefulness of the sequential methods. The separation of the non-optimal options from the best option should also be taken into account.

3. Beta versus Normal Distribution

We expected to find the sequential methods doing better under a normal distribution than under a beta distribution. The results show this difference. There is a big difference between the normal distribution and the beta distribution, and a small difference between the symmetric beta distribution (with μ= 0.5) and the asymmetric beta distribution (with μ=0.1). Figure 17 shows a comparison of the results for these three scenarios:

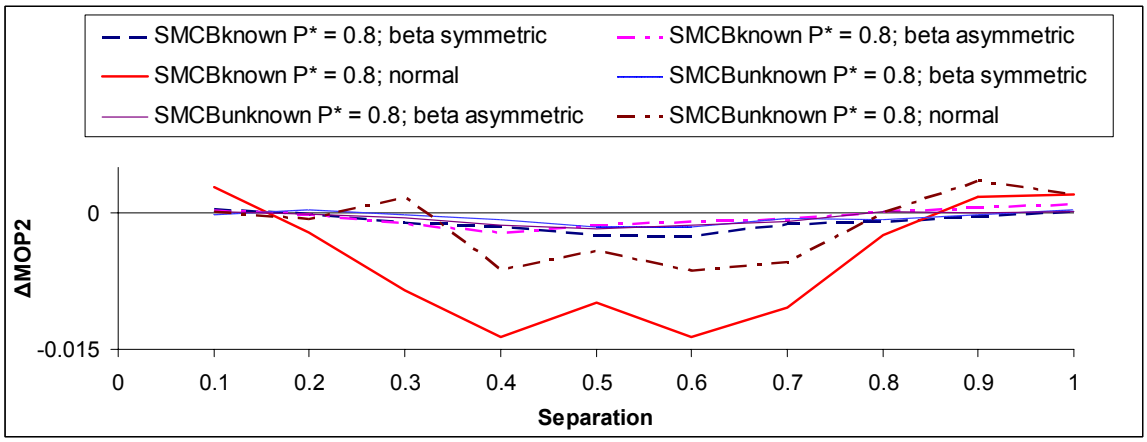

Figure 17. Comparison of results for observations from the normal, symmetric beta and asymmetric beta distributions.
The errors of the methods after subtracting out the base case errors (Δ_{MOP2}). Negative values are favorable for the SMCB methods. Scenarios have 5 options, "half-and-half" separation and a budget of 100 observations.

4. Common Standard Deviation versus Unequal Standard Deviations

The standard deviations of different options can be unequal. We tested the sequential methods in one scenario of unequal standard deviations. The options in this scenario have the same means as the test scenario with nine options, observations from a normal distribution and a "half-and-half" separation, but the standard deviations are unequal – options with a larger mean have a larger standard deviation, such that the ratio μ/σ is the same for all the options.

We find that using unequal standard deviations in the simulations produces inferior results from the $SMCB_{known}$ method. This method assumes a known, common value for the standard deviation, and we consider the use of a value that is not correct for all the options in this test scenario. The $SMCB_{unknown}$ method was less affected by the change of the deviation, and produced results which were sometimes better and sometimes worse, with no obvious difference.

Figure 18 shows the results for a budget of 180 observations:

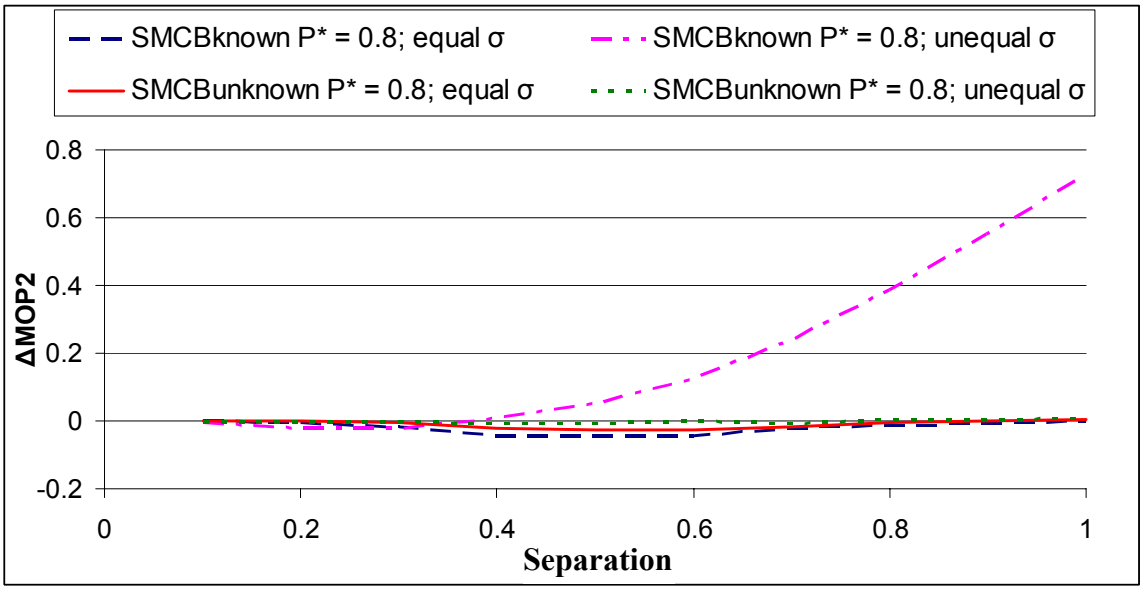

Figure 18. Differences in the relative errors (Δ_{MOP2}) of the sequential methods for equal and unequal variances from the base case

The errors of the methods after subtracting out the base case errors. Negative values are favorable for the sequential methods. Observation from a normal distribution, nine options, "half-and-half" separation, budget of 180 observations.

5. Sensitivity of the $SMCB_{known}$ Method to Misspecification of the Standard Deviation

When using the $SMCB_{known}$ method, it may not be possible to know in advance the exact standard deviation. We use one of the test scenarios to see how the results are affected by assuming a wrong standard deviation. The results show that this method is very sensitive to such inaccuracies.

Using a standard deviation twice as big as the real one, the results were unfavorable especially in scenarios of a small budget. Using a standard deviation half

that of the real one results in an improvement of the results for a small budget, but this advantage quickly turns into a huge disadvantage with a larger budget.

Figure 19 shows the results when assuming the correct standard deviation, half of that and twice of that:

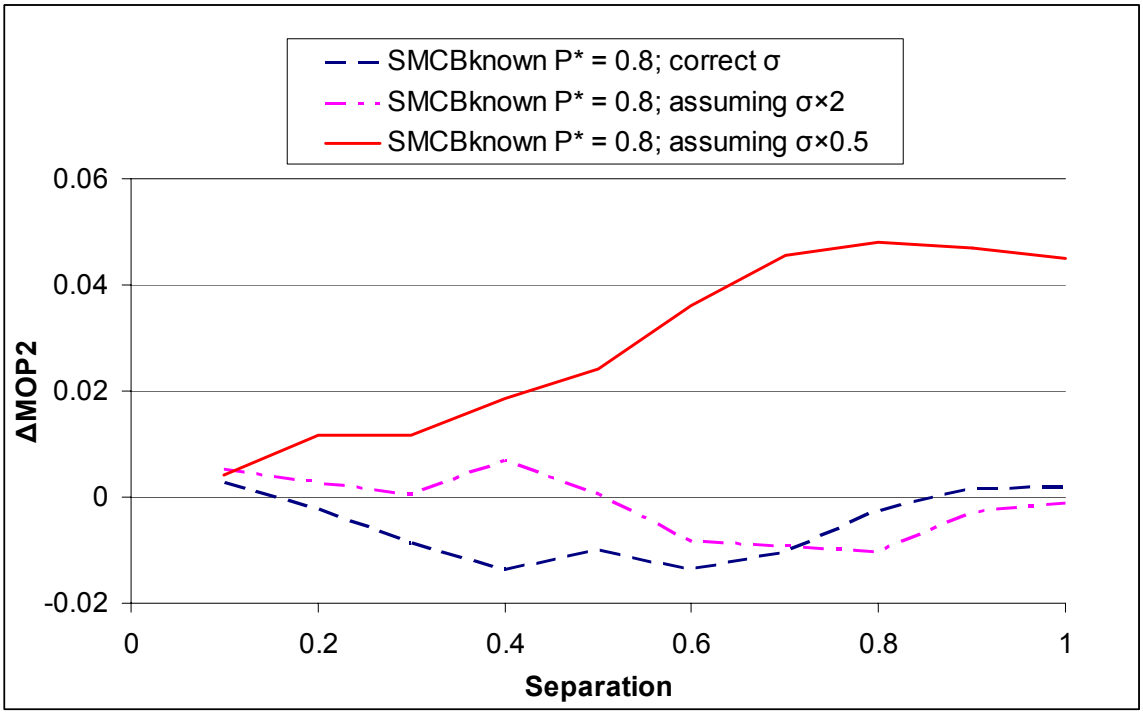

Figure 19. The relative error (Δ_{MOP2}) of the $SMCB_{known}$ method, with wrong standard deviation assumptions.
The errors of the methods after subtracting out the base case errors. Negative values are favorable for the sequential methods. Observations from the normal distribution, five options, "half-and-half" separation, budget of 100 observations.

E. THE MISSILE TRAJECTORY SIMULATION SCENARIOS

We use ten test scenarios for the missile trajectory simulation. These scenarios are described in Chapter III section G. Table 3 shows the performance of the sequential methods compared to the base case in these scenarios, with a budget of 50 and 100 observations:

Scenario	$SMCB_{known}$; P* = 0.8;		$SMCB_{unknown}$; P* = 0.8;	
	budget 50	budget 100	budget 50	budget 100
1	0.0022	0.0106	-0.0005	0.0006
2	-0.0009	0.0027	0	-0.0009
3	0.0024	0.0146	-0.0016	-0.0013
4	0.0137	0.0316	-0.0034	-0.0002
5	0.0005	0.0001	0	0.0002
6	0.002	0.0004	-0.0005	-0.0013
7	0.003	0.0022	-0.0004	-0.0001
8	0.0055	0.001	0	-0.0018
9	0.0047	0.0032	-0.0008	0.0006
10	0.0021	0.0003	-0.001	-0.0008

Table 3. The relative errors (Δ_{MOP2}) of the sequential methods for the missile simulation scenarios.
Negative values (painted green) are favorable to the SMCB methods, positive values (painted orange) are unfavorable.

The $SMCB_{known}$, which assumes a common known standard deviation, usually performs worse than the base case. On the other hand, the $SMCB_{unknown}$ method performs slightly better than the base case.

These test scenarios are arbitrary in nature, and therefore it is difficult to recognize patterns or to say general things about these results. System developers might learn more by testing the sequential methods with pre-existing realistic scenarios. However, these results highlight the sensitivity of the $SMCB_{known}$ method to the distribution, as well as the robustness of the $SMCB_{unknown}$ method.

F. SUMMARY

Compared to the base case, SMCB processes have an advantage in being able to eliminate obviously inferior options early. But this advantage quickly diminishes as the separation between the best option and inferior options decreases. Based on the results, we cannot make a conclusive statement about the advantage of using SMCB processes for moderately separated options.

The sequential methods show potential under some combinations of test scenarios and budgets, but it seems that in order to unlock this potential the system developer needs

to know more about the distributional properties of the simulation data than may be available. We expect the sequential methods to perform better than the base case if there are options that can be identified as inferior and eliminated quickly.

Although our results show that the SMCB processes we consider are not always better than the base case, there may be other processes which perform better. We find that the SMCB processes do not eliminate options fast enough, and they do not take the budget into account when deciding whether or not to eliminate an option. On the other hand, as we noted above, the base case never eliminates any option, no matter how bad it is. However, extending the advantage of sequential methods beyond this case remains a challenging problem.

Other measures of performance, such as the average total number of observations needed to be correct a certain portion of the time, might show a greater advantage for the SMCB methods.

Using the sequential methods may produce an inferior result, compared to the base case. The $SMCB_{known}$ is sensitive to the accuracy of the standard deviation we assume, and a wrong value (especially a smaller one) may generate very big mistakes. On the other hand, the $SMCB_{unknown}$ proved to be much more robust, and is less likely to perform much worse than the base case, even in scenarios which are unfavorable to sequential methods.

V. SUMMARY

A. FUTURE WORK

We tested several sequential selection methods from the statistical literature and changed them according to our needs. In a future work, many other methods can be developed for the elimination of inferior options. We have several ideas for elimination procedures, which we could not test due to a lack of time. Some of these methods use non-parametric techniques which may be less powerful but more widely applicable.

1. Binomial Pairs Comparison

At each stage we compare all possible pairs of options; each such comparison might end with the elimination of one of them. To perform the comparison, we look at the number of times option A is better than option B by comparing the results in the order they were generated in (i.e. first observation of option A versus the first observation of option B and so on).

After counting the number of times the observations of option A are better than those in option B, we assume that we have a binomial test with the null hypothesis that the probability of success is p=50%. According to the number of successes, we can calculate the probability that p is greater than 50%, and the probability that p is less than 50%. If the probability for one of these is high enough, we eliminate one of the options, accordingly.

This method should be calibrated so that the threshold of elimination will take into account the number of live options left, the budget, etc. It is a type of sequential sign test, which bears resemblance to a method described in McPherson and Armitage (1970) for choosing the better option from a pair of options, with a limited budget.

2. Multinomial Comparisons

This method assumes that each option has a probability of being the best at each stage. The null hypothesis is that all the options have an equal probability of $p=1/k$, where k is the number of the remaining live options. At each stage, for each option, we calculate the probability that p is less than $1/k$. If the probability for that is high enough,

we eliminate this option. The method can be changed so that it takes into account the probability of being the best or second best (not only the best), not being the worst or some other combination of this type.

3. Mann-Whitney Comparisons

The Mann-Whitney test uses all observations from a pair of options, ranks them from smallest to largest, and calculates a p-value for testing the null hypothesis that both options have the same distribution against the alternative that one option produces larger observations than the other. Procedures for constructing a SMCB procedure from the Mann-Whitney test can be developed within the framework described by Swanepoel (1977). Investigation of this or similar nonparametric procedures would be worthwhile because they would require minimal distributional assumptions on the data.

4. Normal Distribution, Confidence Interval Overlap

Assume that the observations come from normal distributions with known variances (but unknown means). Since we know the variances, after a few observations we can estimate the mean of each option, and bound it within a confidence interval. As an elimination rule, we can eliminate any option for which the confidence interval of its mean does not overlap with the confidence interval for the best option so far.

The thresholds are set by choosing the bounds (α level) of the confidence interval.

We can also use this approach if we do not know the variances, by estimating them from the observations.

5. Any Chosen Distribution, Using Bayes Theory

If we assume that the observations come from a specific distribution for which we know everything but the mean, we can calculate the likelihood for each observation under different means. Using Bayes theorem we can find the maximum likelihood and construct a posterior distribution of the mean for each option. In order to eliminate an option we look at two options, A and B, with unknown means μ_A and μ_B, respectively. If we can find a value μ', for which the probability p that $\mu_A < \mu'$ and $\mu_B > \mu'$ is high

enough, then it means that these options are separated enough, and therefore we eliminate the inferior one. The calibration of the thresholds is done by choosing probability p.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Hsu, J.C. and Edwards, D.G. (1983). "Sequential Multiple Comparisons With the Best", Journal of the American Statistical Association, volume 78, number 384, Theory and Methods section, 958 - 964.

McPherson, C. K. and Armitage, P. (1970). "Repeated Significance Tests on Accumulating Data when the Null Hypothesis is not True", Journal of the Royal Statistical Society, Ser. A, 134, 15-26.

Swanepoel, J. W. H. (1977). "Nonparametric Elimination Selection Procedures Based on Robust Estimators", South African Statistical Journal, 11, 27-41.

Swanepoel, J. W. H. and Geertsema, J. C. (1976). "Sequential Procedures With Elimination for Selecting the Best of k Normal Populations", South African Statistical Journal, 10, 9-36.

Swanepoel, J. W. H. and Venter, J. H. (1975). "On the Construction of Sequential Selection Procedures", South African Statistical Journal, 9, 103-118.

Swanepoel, J. W. H. and Venter, J. H. (1975). "A Class of Elimination Selection Procedures Based on Ranks", South African Statistical Journal, 9, 119-128.

THIS PAGE INTENTIONALLY LEFT BLANK

CONTACT INFORMATION

The author of this thesis, Gonen Ofer, can be contacted via email at: arnevet@hotmail.co.il, or at arnevet@gmail.com.

The advisor of this thesis, Prof. Robert A. Koyak, can be contacted via email at: rakoyak@nps.edu.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Associate Professor R. Koyak (code OR/Kr)
   Naval Postgraduate School
   Monterey, California

4. Assistant Professor D. Annis (code OR/An)
   Naval Postgraduate School
   Monterey, California