USING SIMULATION TO EXAMINE CUTTING POLICIES FOR A STEEL FIRM

by

Susan Marie Olsen

BS, Pennsylvania State University, 2002

Submitted to the Graduate Faculty of the

School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science in Industrial Engineering

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Susan Marie Olsen

It was defended on

November 24, 2003

And approved by

Jayant Rajgopal, Associate Professor and Graduate Program Director,
Department of Industrial Engineering

Larry Shuman, Professor and Associate Dean for Academic Affairs, School of Engineering

Thesis Advisor: Andrew Schaefer, Assistant Professor, Department of Industrial Engineering

ABSTRACT

USING SIMULATION TO EXAMINE CUTTING
POLICIES FOR A STEEL FIRM

Susan Marie Olsen, MS

University of Pittsburgh, 2003

Minimizing the cost of filling demand is a problem that reaches back to the foundation of operations research. Here we use simulation to investigate various heuristic policies for a one-dimensional, guillotine cutting stock problem with stochastic demand and multiple supply and demand locations. The policies investigated range from a random selection of feasible pieces, to a more strategic search of pieces of a specific type, to a new policy using dual values from a long-range linear program that models a static, deterministic demand environment. We focus on an application in the steel industry and we use real data in our model. We show that simulation can effectively model such a system, and further we exhibit the relative performance of each policy. Our results demonstrate that this new policy provides statistically significant savings over the other policies investigated.

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

A. BACKGROUND 1

B. THE CUTTING STOCK PROBLEM 5

1. Static Demand 5

2. Stochastic Demand 6

3. Classification of CSP 6

a. Dimensionality 6

b. Quantity Measurement 7

c. Shape of Figures 7

d. Assortment 7

e. Availability 7

f. Pattern Restrictions 8

g. Assignment Restrictions 8

h. Objectives 8

i. Status of Information and Variability 8

j. Combined Type 9

C. PROBLEM DEFINITION 9

D. OBJECTIVES 10

E. OVERVIEW 10

II. THE SIMULATION 11

A. SIMULATION STRUCTURE 11

B. ASSUMPTIONS 14

C. PERFORMANCE MEASURES 16

1. Ordered Weight 16

2. Cost 16

3. Inventory Weight 17

4. Scrap Weight 17

D. INPUT DATA 18

III. CUTTING POLICIES 23

A. NOTATION 23

1. Index Sets 23

2. Data 23

3. Variables 24

B. HEURISTIC POLICIES 24

1. Smallest Fit 24

2. Random 25

3. Largest 25

4. Highest Quantity 25

5. Make Highest Quantity 26

6. Multiple 26

7. Extension to Multiple Supply and Demand Locations 27

C. ADELMAN AND NEMHAUSER'S POLICY 27

D. WANG'S POLICY 28

1. Wang's Algorithm 28

a. Objective 28

b. Inventory Limits 29

IV. RESULTS 30

A. SINGLE DEMAND LOCATION, SINGLE SUPPLY LOCATION 30

1. Instance S 30

B. MULTIPLE DEMAND LOCATIONS, MULTIPLE SUPPLY LOCATIONS 38

1. Instance A 41

2. Instance B 49

3. Instance C 54

4. Instance D 59

5. Instance E 63

C. SUMMARY 68

V. CONCLUSIONS AND EXTENSIONS 71

BIBLIOGRAPHY 73

LIST OF TABLES

Table No.		Page
1	Instance S: 95% Confidence Intervals for Mean Cost by Policy	34
2	Instance S: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)	34
3	Instance A: 95% Confidence Intervals for the Mean Cost by Policy	48
4	Instance A: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)	49
5	Instance B: 95% Confidence Intervals for Mean Daily Cost	53
6	Instance B: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)	54
7	Instance C: 95% Confidence Intervals for Mean Cost for Each Policy	58
8	Instance C: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)	58
9	Instance D: 95% Confidence Intervals for Mean Cost	62
10	Instance D: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)	62
11	Instance E: 95% Confidence Intervals for Mean Cost	67
12	Instance E: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)	67
13	Relative Performance of Daily Cost Across all Instances	68
14	Graphical View of Daily Cost by Instance	69

15 Relative Performance of Daily Scrap Weight Across all Instances 69

16 Relative Performance of Daily Inventory Weight Across all Instances 70

LIST OF FIGURES

Figure No. Page

1 Steel in Coil Form [14] 2

2 Steel in Sheet Form [14] 2

3 Example of Guillotine Cutting Restriction 3

4 Simulation Flowchart 13

5 Instance S: Long-Run Ordered Weight by Day 20

6 Instance A: Long-Run Ordered Weight by Day 21

7 Instance S: Average Daily Cost for Each Policy 31

8 Instance S: Average Daily Cost Over All Replications by Policy 32

9 Instance S: 95% Confidence Intervals for Mean Cost by Policy 33

10 Instance S: Average Daily Ordered Weight by Replication 35

11 Instance S: Average Daily Inventory Weight by Policy 36

12 Instance S: Average Daily Scrap Weight by Policy 37

13 Layout of Sample Problem A 39

14 Layout of Sample Problem B 39

15 Layout of Sample Problem C 40

16 Layout of Sample Problem D 40

17 Layout of Sample Problem E 41

18 Instance A: Average Daily Ordered Weight 42

19 Instance A: Average Daily Inventory Weight by Policy 43

20 Instance A: Average Daily Scrap Weight by Policy 44

21 Instance A: Average Daily Cost by Policy 45

22 Instance A: Average Daily Cost by Policy over All Replications 46

23 Instance A: 95% Confidence Intervals on the Average Daily Cost 47

24 Instance B: Average Daily Cost for Each Policy 50

25 Instance B: Average Daily Cost by Policy over All Replications 51

26 Instance B: 95% Confidence Intervals for Mean Daily Cost for Each Policy 52

27 Instance C: Average Daily Cost by Policy 55

28 Instance C: Average Daily Cost by Policy over All Replications 56

29 Instance C: 95% Confidence Intervals for Mean Cost for Each Policy 57

30 Instance D: Average Daily Cost by Policy 59

31 Instance D: Average Daily Cost by Policy over All Replications 60

32 Instance D: 95% Confidence Intervals of Mean Cost for Each Policy 61

33 Instance E: Average Daily Cost by Policy 64

34 Instance E: Average Daily Cost by Policy Over All Replications 65

35 Instance E: 95% Confidence Intervals for Mean Cost for Each Policy 66

CHAPTER I

INTRODUCTION

A. BACKGROUND

In 1991, the United States steel industry sold over $27.3 billion worth of material and employed over 180,000 people in this country alone [12]. In September 2003, over 78.1 million metric tons of steel were produced throughout the world, an increase of 3.7% over September 2002 [11].

Sheet steel comprises the largest portion of the steel business[5]. Customers to this segment include the automotive, appliance, construction, and electrical industries, as well as the agricultural and industrial equipment industry. The actual products required by these customers vary greatly, from stainless to galvanized, to hot-rolled steel. In addition, the product dimensions, length, width, and thickness (gage), vary from order to order as well.

Although steel is desired in all shapes and sizes, it is simply not cost efficient for a mill to produce small orders (less than a full coil) to the exact dimensions requested. Instead, steel is mass produced in large coils. Note, these coils are essentially one long sheet of steel of a given width. These coils, pictured in Figure 1, are produced in many different widths.

Figure 1: Steel in Coil Form [14]

Frequently, steel is not actually used in coil form however, and is instead needed in sheets. Since a coil is essentially one extremely long sheet, in order to create several shorter sheets from a single coil, the steel is unrolled and cuts are made width-wise across the steel. The result is steel in sheet form as pictured in Figure 2.

Figure 2: Steel in Sheet Form [14]

Obviously, not every customer desires sheets of the same width. Therefore, coils of varying widths are needed. While mills are able to produce coils in varying widths, it is not efficient to produce an entire coil of a specified width if there is only a demand for a short length of that particular width. Instead, that demanded piece should be cut from a larger, available piece.

Coils can also be cut length-wise to create two narrower coils. This is done to satisfy demand for widths that are either too narrow for mills to produce, or in the case that the proper width of coil is unavailable for immediate use. However, it is important to note that all cuts are made through the entire length (or width) of the steel, meaning no “L-shaped” pieces can be created (nor can they be resultant). This restriction is due to machine capabilities and is commonly referred to as a guillotine cut restriction. Figure 3 provides an example of 2-dimensional guillotine cuts.

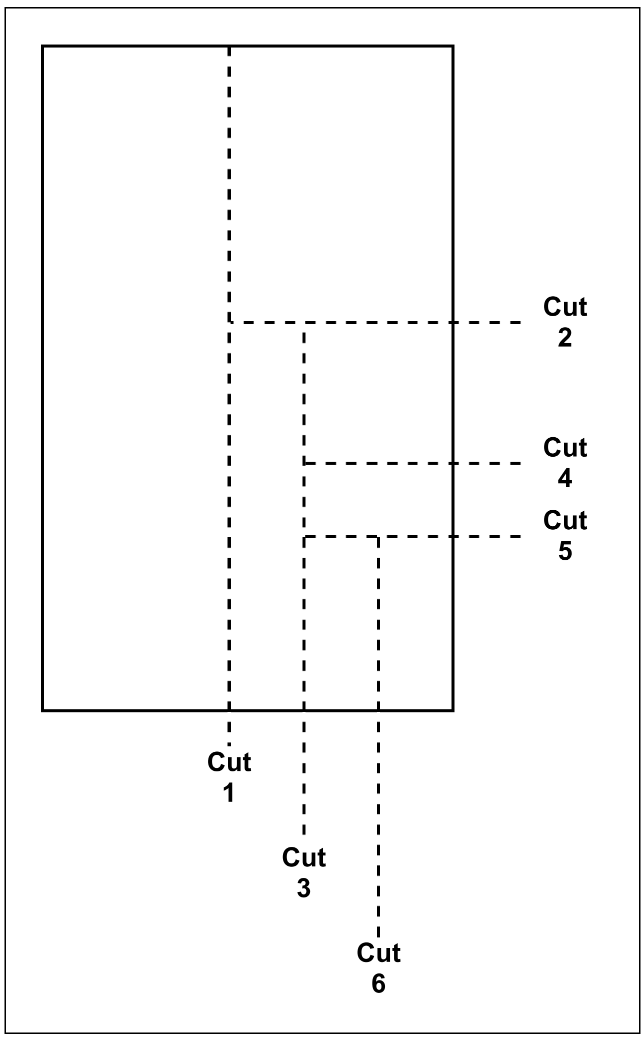

Figure 3: Example of Guillotine Cutting Restriction

Although new technologies are developing within the industry everyday, most steel mills use similar equipment and have comparable capabilities. Thus, if one steel supply company is unable to gain a significant edge over a competitor due to technological or manufacturing capabilities, they must look elsewhere within their business for such an advantage.

One such area of advantage comes in the form of a careful business plan. If one company is able to fully understand their customers' demand, they have the basis to develop a better plan to handle that demand. But, a complete and accurate forecast is only helpful if one is able to take that information and use it to meticulously plan their demand fulfillment. One must decide what steel to produce when and where, and what width source coil to produce it to. Successful steel firms are able to meet their customers' demands while producing and scrapping the least amount of steel. Although it may seem complicated enough to determine how to best cut one piece of steel from another, the situation is further complicated by the availability of various sized coils, the costs and capacities of processing plants, the shipping costs associated with delivering the finished product to the customer, and the uncertainty of the actual demand. However, we will show that a new policy developed by doctoral candidate Zhouyan Wang [15] is able to fully consider all of these factors and return information such that any planner could develop an optimal policy for handling demand.

We consider the problem one-dimensionally, meaning that each piece of steel is of varying widths, yet their lengths remain constant (120 feet). Examining the problem in this way allows us to demonstrate the actual costs associated with each of the various policies most apparently. One could imagine that the cost differentials between policies will only expand when the cutting policies are applied to other dimensions, but we will not complicate the policy with such expansion at this time.

B. THE CUTTING STOCK PROBLEM (CSP)

1. Static Demand

The Cutting Stock Problem (CSP) is one of the earliest problems pondered in the field of Operations Research. The first known CSP was formulated by the Russian economist Kantorovich in 1939 [8]. The problem has wide applicability and strong economic implications and is therefore investigated in many industries, sometimes under different names. For example, in the paper industry the CSP is referred to as the deckling problem [6]. In all cases, the CSP remains an issue, because it is nearly impossible to build and solve an appropriate integer program to optimality [13]. Because of this issue, most researchers focus their attention on either heuristics or linear programming relaxations [13].

The approach that a researcher decides to take is largely based on the specific application with which he is working. For example, in one-dimensional problems, a frequent consideration is to minimize trim loss. That is, one minimizes the amount of unusable material resulting from cutting the raw material to meet demand. However, this wasted material is not the only cost worth consideration. Other researchers focus their attention on minimizing the cutting pattern changes. Haessler (1975) even provides a formulation combining the two objectives. Another cost worth consideration is the transportation cost associated with moving the final product from the supply depot to the demand point.

Adelman and Nemhauser (1999) consider a problem that is very similar to the one we examine here. However, their model only accounts for a single supply and a single demand location. Their work provides the motivation for our investigation and will be discussed further in Section 3.3.

2. Stochastic Demand

Although stochastic demand may be a reality in many systems similar to the one we discuss here, few people have explored it in their model formulations. An exception is Krichagina, et al. (1996) who did consider stochastic demand in their model of a paper plant. However, their model did not consider remnant inventories and it also only considered a single supply/demand location.

3. Classification of CSP

Dyckhoff (1990) published a typology of CSP's. He qualified each problem in seven ways. Here, we discuss how our problem fits each category.

a. Dimensionality

Dyckhoff defined dimensionality as, "the minimum number of dimensions of real numbers necessary to describe the geometry of patterns." Here we will work in one-dimension. However, the dimensionality aspect is actually more complicated than that. Since our problem also has the specific restriction of requiring guillotine cuts, he considers it to be '1 + 1-dimensional'.

b. Quantity Measurement

That is, “the way of measuring the quantity of the large objects and of the small items, respectively.” He considered two classes: discrete or integer measurement, and continuous or fractional measurement. Our problem falls into the discrete category.

c. Shape of the Figures

Different applications may require shapes of a wide variety. Here we are only considering rectangles. Therefore, his classification puts our problem into the different (regular) forms category. This category allows us to have rectangles with distinct ratios of width to length.

d. Assortment

“The assortment is given by the shapes and the number of permitted figures.” Here again, we are dealing with rectangles of specified dimensions, but it is not a necessary condition that only congruent rectangles may be cut from the source rectangle. In fact, that is not likely to be optimal due to trim loss. There is no limit on the number of figures that may be cut from a source rectangle. The source is exhausted only when its dimensions fall into the scrap category. That is, the dimensions of the remnant are smaller than the smallest possible order size.

e. Availability

We cover availability in two ways. First, we impose a limit on the number of source pieces arriving daily, and second, we limit the number of pieces that a warehouse can store in inventory. We do not impose a sequence or order, for our material, steel, generally has a long shelf life (approximately 5 years).

f. Pattern Restrictions

One dimensionally, our problem has few restrictions. There is no minimum distance required between finished parts, no orientation restrictions, nor any frequency restrictions. However, we are dealing with the restriction that all of our cuts will be guillotine cuts. That is, whenever a cut is made, it must continue across the entire width of the sheet.

g. Assignment Restrictions

We assign one order to one stock piece at a time and we do not aggregate.

h. Objectives

Here, we will minimize the cost of the raw material used, less the scrap refund recovered, plus the cost of transporting the steel from the warehouse to the customer's demand location. This is what we assume to be the total cost of the model.

i. Status of Information and Variability

Here our demand is stochastic in nature. Although we cannot guarantee the dimensions of requested materials, we are allowed a 10% tolerance on our orders. Therefore, we assume that the dimension we produce is the dimension we ordered.

j. Combined Type

Dyckhoff then simplifies all of these classifications into combined types. These types consist of the four main characteristics of a problem: dimensionality, kind of assignment, assortment of large objects, and the assortment of small items. He then assigns symbols to each possible classification within each characteristic. Then, according to his notion, our problem can be considered a 1/B/D/R.

C. PROBLEM DEFINITION

Our simulation is modeled on the steel industry, and Straightline, a division of U.S. Steel, provided actual data for this simulation. We utilize this data to better represent an actual order-processing system. However, this data is not used directly in our system, but rather it is used to determine the underlying distribution of stochastic demand. The goal of the project is to analyze the current process for planning steel orders and to compare it with a new method [15].

Straightline's vision is to, "revolutionize the steel business" by putting the customer in control [14]. The company provides its customers an interactive website and ultimately, the opportunity to purchase steel in any size and shape they desire. Since the livelihood of the company is contingent on satisfying its customers' requirements efficiently and cost-effectively, Straightline is always seeking way to improve upon its current processes.

Straightline's goals are not uncommon in the business world. We seek to investigate these goals by modeling the order processing procedures using a variety of heuristic policies. Among these policies there is a new policy that uses dual values from a linear program to determine the inherent value of remnant pieces. Interestingly, this linear program considers only static, deterministic demand, while our model will consider stochastic demand.

D. OBJECTIVES

The objectives of this research are listed below:

1. To investigate current cutting policies used in the steel industry
2. To research alternate methods that could be applied to the situation.
3. To build a simulation to compare policies and to demonstrate the effects of each.

E. OVERVIEW

In Chapter 2 we will discuss the simulation structure and inputs. Then in Chapter 3 we present the cutting policies that we investigated, with particular interest in the policies of Adelman and Nemhauser [2] and Wang [15]. The results of our investigation are summarized in Chapter 4. Finally, conclusions drawn on our results, as well as an outline of future directions for this research are presented in Chapter 5.

CHAPTER II

THE SIMULATION

A. SIMULATION STRUCTURE

There are several ways to investigate the effectiveness of a cutting policy. Certainly, simply implementing a new cutting policy may be a bit rash. After all, if something went wrong, the company could suffer complete financial ruin. Instead, the new policy could be implemented in stages; but once again, there could potentially be some major pitfalls. The old policy and the new policy may be formulated very differently. This would lead to different priorities in supply and demand and could ultimately be very destructive.

In order to observe our plan in action, without threatening Straightline's financial well-being, we elected to build a simulation. This simulation allows us to model the real-world situation, using a variety of cutting policies, so that we may compare, contrast, and ultimately determine the optimal policy.

The simulation model is built in C++. This is a convenient choice for the modelers both in the flexibility of the language and the familiarity of the modeling team with the language. Further, the model incorporates discretized real data into the incoming demand, and roughly estimates supply and demand depots throughout the United States.

Simulation is a powerful tool that provides an immense amount of information. We can easily run the model for any length of time and any number of locations. Then, we can investigate the model outputs to determine which policy performs optimally under which circumstances, and also whether the policies' performances are significantly different. The replication data allows us to investigate whether there is a significant difference in the results between runs.

The simulation model was built in many steps, starting with the very basic, and building up to the rather complex. This allowed us to constantly verify that the model was performing as expected in each circumstance, without too much difficulty. Also, the code was reviewed by one of the team members to assure that everything was proper. Finally, our contact at Straightline is able to validate that our results correspond to likely events.

A schematic of the basic simulation operation is displayed in Figure 4.

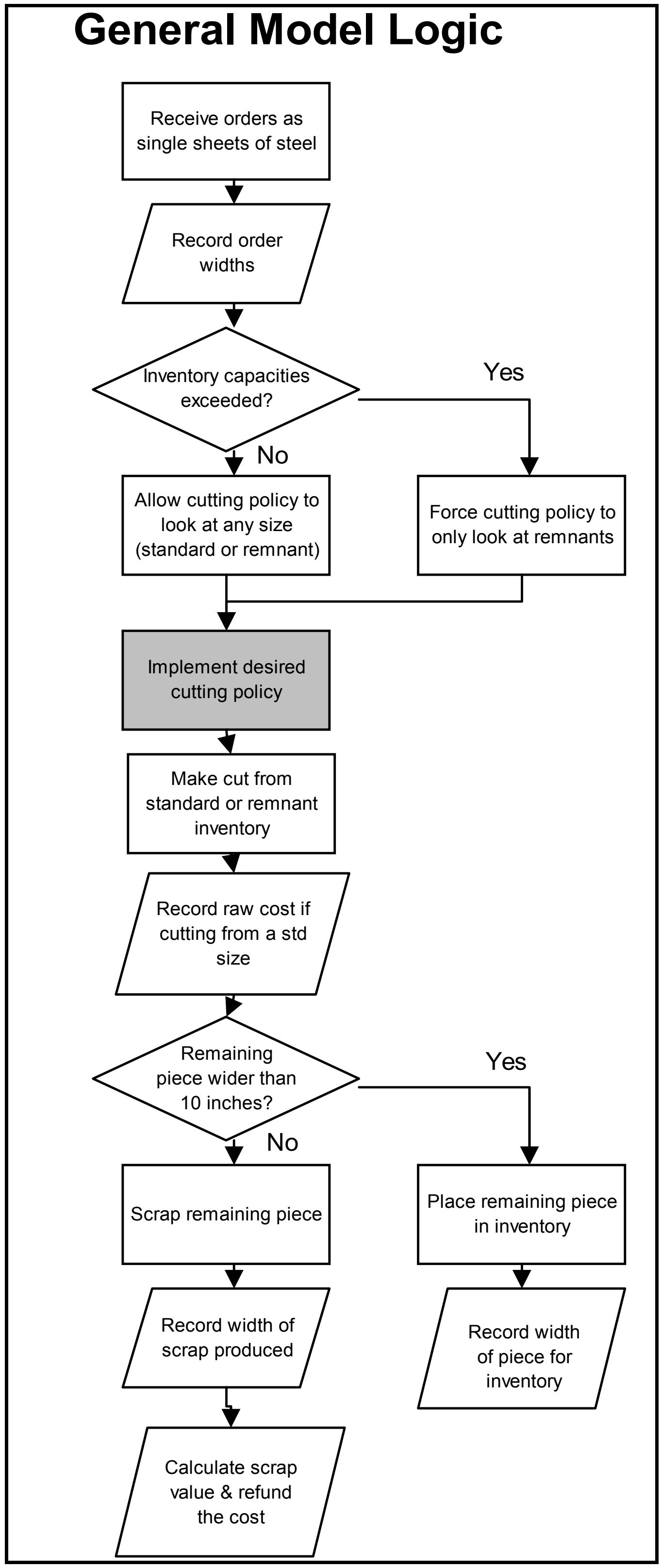

Figure 4: Simulation Flowchart

B. ASSUMPTIONS

Every attempt is made to ensure that the simulation model resembles real-world activity. However, programming time was limited and for simplicity, not every single variable encountered on a daily basis can be included. The main objective of our study is to investigate cutting policies, so we have designed our model to show how these policies most affect the real-world system.

For example, although it would be simple to create a demand set that is steady or fluctuates little, this is not representative of the real world. For our model, we discretized actual data so that it shows the stark contrasts in demand for pieces of different sizes. However, we do not simply implement this data as though it were the actual demand. Instead we recognize the stochastic nature of actual demand and therefore we use an exponential distribution based on this discretized data to represent the actual data.

Also, it is important to investigate the relative availability of stock sizes versus ordered sizes. It is obviously easier to fulfill demand efficiently when every size is already available to you. Again, we turned to the real-world system to investigate the proportion of raw sizes available versus the number of sizes that orders are filled for. From this information we determined a set of “standard sizes”. All orders were then cut either directly from these standard sizes, or indirectly from them by using the remnant remaining from an earlier cutting operation.

Further, there are multiple ways for a company to handle its inventory limits, and we wanted to demonstrate the method that is the best balance of simplicity and effectiveness. Therefore, we investigated many different ways of approaching the issue and tested each of them. After this investigation, we decided that the best method was to limit each policy based on the current model. After several rounds of testing, we determined that the optimal limit was 10 sheets of steel for each size. In many cases, limiting the inventory actually increased the cutting policies' efficiency.

Another inventory issue that required attention was the issue of inventory availability. Although it would be simple to assume that whenever a piece of steel was requested it was immediately available, that assumption is not very realistic. In reality, we must decide in advance what sizes to order so that we have the appropriate sized pieces on-hand when needed. Wang's method not only determines an optimal cutting policy, but it also outputs optimal consumption rates for raw material. These consumption rates can then be used to determine optimal raw size ordering rates. Then, those raw sizes, in their predetermined quantities, arrive into the system daily. However, we did not determine such ordering rates for the other policies and instead assumed that the other policies had unlimited access to all standard sizes at all times. Although this may introduce bias into the system, this bias is actually supportive of the other policies and detrimental to Wang's policy. Additionally, we show the effect of this restriction by giving results with and without it in place. Our computational experiments show that Wang's policy still appears to be optimal, despite this bias.

Assumptions were also made regarding transportation between cities. For simplicity, we decided that we would charge a consistent per-mile per-inch transportation rate across the entire country.

Demand locations were randomly selected, however, steel supply locations were located arbitrarily rather than randomly, because random placement could easily place a steel mill in a completely unlikely and unfavorable location. The steel supply locations were held consistent between various model runs to investigate the effectiveness of the model under different demand conditions.

C. PERFORMANCE MEASURES

In order to analyze the performance of the model, one must determine certain performance measures to collect. These measures are collected each day that the simulation is run and then are also summarized per replication.

1. Ordered Weight

This measure tracks the total amount of demanded weight that the model must handle each day. Ordered weight is dependent upon both the number of orders received each day and the dimensions of those orders.

2. Cost

The cost tracked here is the cost to fulfill the incoming demand, both in the raw material cost and in the shipping cost, less some refund for scrapping unusable pieces. Raw material cost is calculated whenever a standard size is cut. Remnant pieces can then be used at no cost. One objective may be to minimize the cost that the steel firm must pay to fulfill all of its customers' demands.

3. Inventory Weight

This measure tracks the total amount of inventory that the model is holding on a daily basis. This includes the entire inventory across all of the supply depots. Inventory contains steel that we have already paid for but have not yet used, thus, it is generally best to keep inventory to a minimum.

4. Scrap Weight

Our final performance measure, scrap weight, tracks the amount of steel that has been scrapped by the model. By definition, a piece is scrap if it is smaller than the smallest possible demanded size. Also, our model forces a piece to be scrapped if it violates our inventory restrictions (it causes us to have more inventory than we can handle). There is no clear way to optimize the system based on scrap weight, which makes it an interesting measure. A high scrap weight may indicate that the standard sizes purchased were too large or that the system more effectively removed unnecessary pieces. Then, a low scrap weight may result from a system where the proper standard sizes were used or from a system where more pieces were held in inventory instead of being scrapped.

D. INPUT DATA

Model inputs are frequently subject to randomness. There are several inputs into our simulation. These include number of orders received per day and the dimensions of those orders. There are three main ways to handle this randomness in a simulation.

1. The actual data values themselves could be used directly in the simulation.
2. The data values can be used to define empirical distribution functions.
3. The data could be fit to a theoretical distribution.

Option 1 was not plausible for our simulation as we did not have enough data for all of our runs. Further, the data that we do have is historical and would therefore be limiting in our prediction of future events.

Option 2 is not desirable due to the limited scope of our data. Creating an empirical distribution would require significant confidence that our data is very representative of actual events. However, if there are any irregularities in the data, these will also come forth in the empirical distribution, outputting non-representative values.

Option 3, fitting theoretical distributions, was the best option for each of our sources of randomness. We determined that the best fit for the distribution of the number of incoming orders (number of sheets) per day, and the width of the orders was the exponential distribution. The exponential distribution has the convenient property of memorylessness. This is important here because we assume that the quantity and dimension of incoming orders is not dependent on previous orders. This assumption is not unrealistic, for Straightline has a wide base of customers and does not receive the majority of the orders from the same few customers. Thus, what one customer orders in Texas has no bearing on what another orders in New York. Similarly, the timing of one customer's orders has virtually no effect on the timing of another customer's orders.

Based on the data we received from Straightline, we were unable to determine the number of incoming orders per day precisely, but rather we were able to examine the orders that they considered on a daily basis. This is an important distinction, because the planner at Straightline has the opportunity to delay a decision on a particular order to the next day if he desires. This flexibility is not accounted for in our model due to our lack of data and our desire to maintain simplicity whenever possible.

Although we did use the data provided to investigate the arrivals of orders, we could not use it directly for two reasons. First, we did not have enough data to input to run our simulation for the length of time desired. Second, we are aware that the order sizes requested vary over time. Therefore we decided to use the means of the data as inputs into the exponential distribution. Although the exponential distribution presents the possibility of generating arbitrarily large data values, we were able to avoid this situation by setting truncation values reasonable levels.

As our simulation considers many random variables per day, it is very important to carefully select a random number generator to appropriately handle this situation. We elected to use a Mersenne Twister pseudo-random number generator [10]. The primary reasons for selecting this particular generator were for its speed and period. Also, this generator was also chosen for its efficient use of memory.

There were also several other important inputs to determine, including the run-length, warm-up period and the number of replications to be run. As our model is a non-terminating system, we implemented the replication/deletion technique for determining warm-up period, run-length, and the number of replications.

The replication/deletion method is used to remove bias from a system to allow for a reduction in variability and ultimately, to form confidence intervals on various performance measures [10]. This is achieved by removing initial transient data and then performing independent replications. In each replication (we use 10), the model is initialized and the transient data is deleted in the same way.

First, we made one very long run for each of our instances. Then, we analyzed the total weight ordered each day to determine the length of the initial transient data. Figure 5 shows this graph for Instance S, a single supply/single demand location instance, and Figure 6 shows this graph for Instance A, a multiple supply/multiple demand locations instance.

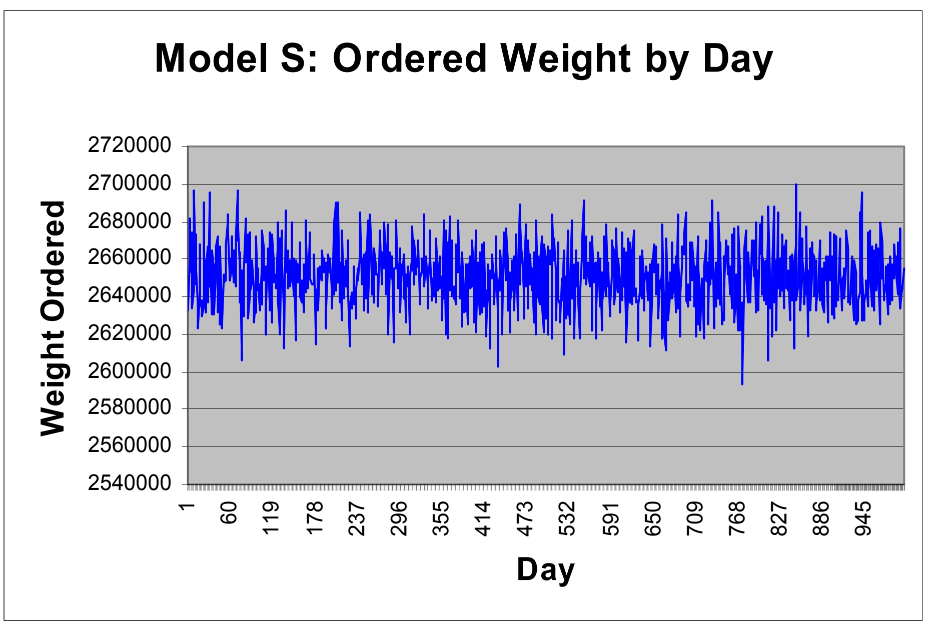

Figure 5: Instance S: Long-Run Ordered Weight by Day

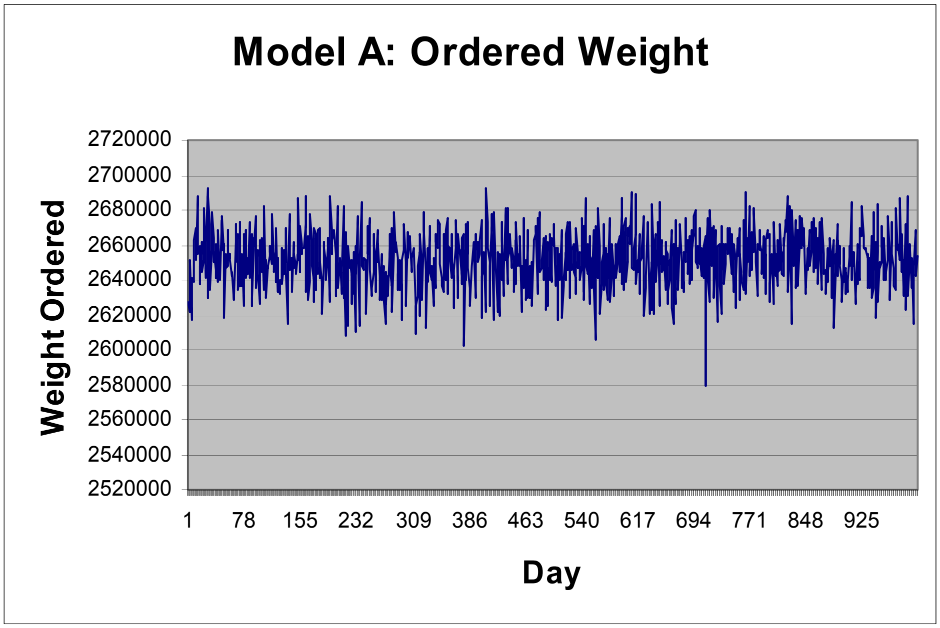

Figure 6: Instance A: Long-Run Ordered Weight by Day

In order to successfully implement the replication/deletion methodology, we first determined an adequate warm-up period. Notice that neither graph indicates a clear initial transient period. In fact, we believe that the actual warm-up period is only a fraction of a day. The only factor that appears periodically is the inventory capacity situation. That is, at the very beginning of the first day of the simulation, there are no remnants in inventory. Thus, all initial orders are cut from standard sizes. However, we limit the total size of the inventory to fewer than 500 pieces and yet we handle an average of 20,000 pieces per day. With a run length of several days, this initial transient period is quickly dominated. However, in order to ensure model accuracy, we determined a warm-up period of one day.

The replication/deletion method states that as a rough rule, the run-length should be at least ten times longer than the warm-up period whenever possible [3]. Although this indicates that a run-length of 10 days would be adequate, we decided to use a run-length of 20 days. Then, each day is considered a batch, and our first batch is deleted. Confidence intervals may then be calculated based on the remaining 19 batches to determine the long-run mean of each performance measure.

Finally, we considered the total number of replications required. Fortunately, the model run-time did not restrict us in such a determination. We started with 10 replications and then checked to see if that number was adequate. In fact, for each of our instances, 10 replications were enough to ensure that with 95% confidence our half-width was within 0.2% of our sample mean for all of our performance measures.

CHAPTER III

CUTTING POLICIES

A. NOTATION

1. Index Sets

I = index set of facility locations.

J = index set of demand locations.

K = set of all possible sizes (including scrap) that could be generated (across all facilities).

$D \subset K$ = subset of all sizes that are demanded at one or more $j \in J$.

$S \subset K$ = subset of all raw stock sizes that are processed at one or more $i \in I$.

2. Data

$\lambda_{j,k}$ = demand rate at location $j \in J$ for a product of size $k \in D$.

$c_{i,j,k}$ = cost of transporting one unit of size $k \in D$ from location $i \in I$ to location $j \in J$.

$a_{i,k}$ = unit cost of raw stock size $k \in S$ at location $i \in I$.

σ_i = unit salvage value of one unit of product at facility $i \in I$.

3. Variables

$x_{i,j,k}$ = rate of shipment of units of size k from facility location i $\in$ I to demand location j $\in$ J. Define $x_{i,j,k} = 0$ if a unit of size k either cannot be generated at location i or is not required at location j.

$y_{m-n,i}$ = rate of generation at facility location i $\in$ I, of units of size (m-n) that are obtained by cutting size m down to size n.

$r_{i,k}$ = rate of replenishment of raw units of size k $\in$ S at facility location i $\in$ I.

$s_{i,k}$ = rate of production of scrap of size k at facility location i $\in$ I.

B. HEURISTIC POLICIES

1. Smallest Fit

1. Find the available piece that is equal in size to the desired piece, and cut the order from it.

$$\text{Arg } \{m\text{-}n \mid m\text{-}n = 0\}$$

2. If the actual size is not available, cut from the smallest non-standard size piece that is large enough to fulfill the demand.

$$\text{Arg Min } \{m\text{-}n \mid m\text{-}n > 0,\ m \text{ non-standard size}\}$$

3. If no such non-standard size piece exists, cut from the smallest standard size piece that is large enough to fulfill the demand.

$$\text{Arg Min } \{m\text{-}n \mid m\text{-}n > 0,\ m \text{ standard size}\}$$

2. Random

Find all pieces equal or larger than the ordered size and randomly select one to cut the order from.

$$\text{Arg Random } \{m \mid m \geq n\}$$

3. Largest

1. Cut from the largest available acceptable non-standard piece.

$$\text{Arg Max}\{m\text{-}n \mid m\text{-}n \geq 0,\ m \text{ non-standard}\}$$

2. If no non-standard pieces of adequate size exist, cut from the smallest adequate standard piece.

$$\text{Arg Min}\{m\text{-}n \mid m\text{-}n \geq 0,\ m \text{ standard}\}$$

4. Highest Quantity

Find all acceptable pieces (greater or equal to order size), then compare their stock quantities. Cut from the size that has the greatest number of pieces in inventory. Break ties by choosing the smallest stock size.

$$\text{Arg Max}\{\ (y_{m-n,i}) \mid m\text{-}n \geq 0\}$$

5. Make Highest Quantity

Find all pieces available (greater or equal to order size), then determine the remnant size that would remain if the order was cut from that stock piece. Choose the stock piece that would generate the most common stock size. Break ties by choosing the smallest stock size.

$$\text{Arg Max}\{\, (y_{m\text{-}n,i}) \mid m\text{-}n \geq 0\}$$

6. Multiple

1. First, if an inventory piece of the exact size of the demanded piece exists, use that piece to fill the demand.

   $$\{m\text{-}n \mid m\text{-}n = 0\}$$

2. If such a piece is not available, look for a non-standard sized piece that would create a resultant piece that is smaller than the smallest acceptable order size (scrap). Of these pieces, cut from the one that will result in the smallest piece of scrap.

   $$\text{Arg Min } \{m\text{-}n \mid m\text{-}n < \min (n),\ m \text{ non-standard}\}$$

3. If no such pieces exist, cut from the largest non-standard piece in stock.

   $$\text{Arg Max}\{m\text{-}n \mid m\text{-}n \geq 0,\ m \text{ non-standard}\}$$

4. Finally, if no non-standard pieces in stock are large enough to fill the order from, reconsider steps 2 and 3 for the standard raw sizes.

7. Extension to Multiple Supply and Demand Locations

There are many different ways to extend these policies to account for multiple supply and demand locations. We consider this extension in a simple, yet realistic way. That is, we first find the supply location that is nearest to our demand. If there is a tie, we break it randomly. Then, we follow one of the above policies to fill the demand from that nearest facility's supply.

C. ADELMAN AND NEMHAUSER'S POLICY

A closely related problem in the fiber optics industry was researched by Adelman and Nemhauser (1999). There, lengths of fiber optics were cut from standard sizes and/or remnants to satisfy static demand. Adelman and Nemhauser built a linear programming model of the system and used the resulting duals as inherent values for the various lengths of fiber optics. Their cutting policy entailed cutting from the piece that created the smallest reduction in value. They built a simulation and were able to demonstrate that their policy presented great savings over traditional heuristic policies.

However, Adelman and Nemhauser's policy is not directly applicable to this problem. Unusable lengths retain no value in the fiber optics industry, thus no scrap values were considered. Yet, in the steel industry, scrap remnants are recycled and therefore represent significant value.

Further, Adelman and Nemhauser considered only a single supply, single demand set-up. We, however, are managing a network of supply and demand points and must encompass this complexity into our model.

D. WANG'S POLICY

Wang (2003) researched the work of Adelman and Nemhauser and tailored it to include the extra complexity of our system, stochastic demand and remnant scrap value.

1. Wang's Algorithm

Calculate dual prices for all feasible stock sizes (original stock sizes, the sizes that may result after cutting different sized orders from the stock sizes, and the sizes that would result from those remnants). Also, the scrap values of the unusable remnants (narrower than the smallest demanded size) play a role in the dual price calculations, and the transportation costs. His program for calculating duals also takes into account the daily average demand.

a. Objective

Selecting a piece to cut from using this method requires a calculation of the reduction in value of the piece. That is, for all stock sizes available that are greater than or equal to the order size, subtract the dual price value of the resultant piece from the current value of the stock piece. Choose to cut from the stock piece that will result in the smallest reduction in value.

$$\text{Arg Min}\{\eta_{m-n} - \eta_m\}$$

<u>Note</u>: Ties are frequent with this policy. Several tie-breaking policies were implemented and compared. They are as follows:

Cut from the next larger piece (smallest fit)

Cut from a randomly selected piece (random)

Cut from the piece with the highest quantity in inventory (highest quantity)

b. Inventory Limits

Wang's linear program results include optimal raw size consumption rates. We investigate the transferability of this information to our model. That is, we restrict incoming inventory availability to the optimal consumption rates that Wang's output provides (plus a safety factor of 10%). We compare the performance of Wang's model under these conditions with the performance of Wang's model without these incoming inventory restrictions.

CHAPTER IV

RESULTS

All policies are described in the previous section. Wang's policies were tested with each of three tie-breaking rules (shown in parentheses) as well as with and without incoming inventory restrictions (limits).

A. SINGLE SUPPLY, SINGLE DEMAND LOCATION

1. Instance S

Instance S is a single supply/single demand location instance. For this instance, there is only one facility from which demand may be fulfilled, and thus, shipping costs can be ignored. We ran this instance for 20 days (with a 1 day warm-up) and for 10 replications.

Figure 7 shows the daily cost averages per replication for each policy. This graph shows the relative performance of each policy, however, we are unable to draw any statistical conclusions based on this graph alone. Note that the Random policy has been removed from this graph as this policy performs far worse than all others. Removing it from the graph allows one to more closely examine the relationship between the other policies.

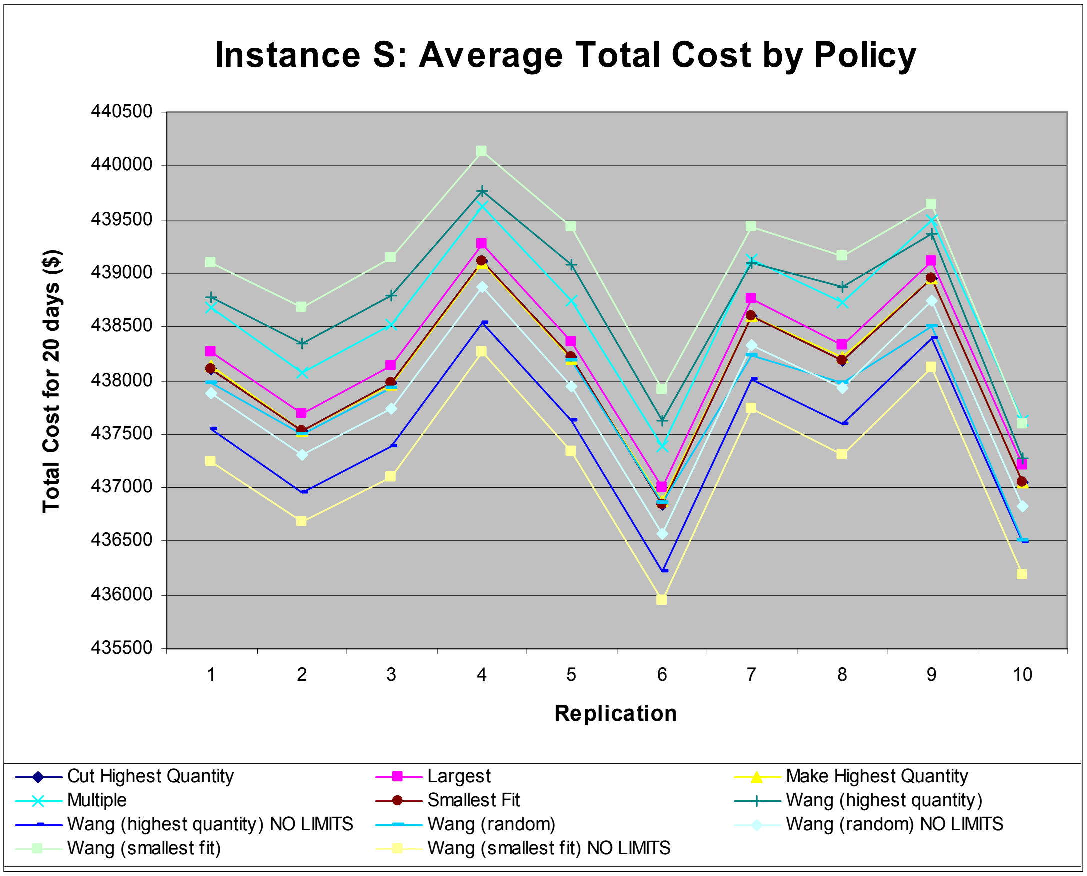

Figure 7: Instance S: Average Daily Cost for Each Policy

Figure 8 summarizes the average cost of each policy over all replications. Once again, this graph alone is not adequate for drawing statistical conclusions. Therefore, we created Figure 9, a graph of confidence intervals for the average cost by policy. Table 1 explicitly lists the interval values. One can determine the statistical independence of the policies by evaluating these intervals. If two intervals are wholly independent (not overlapping), the policies are determined to be statistically different.

Our 95% confidence intervals are not inclusive of all replication means, and we are unable to conclusively give a reason for that at this time. We suspect that repeated replications and further statistical analysis would reveal that one of our replications is drawing on a bad seed value.

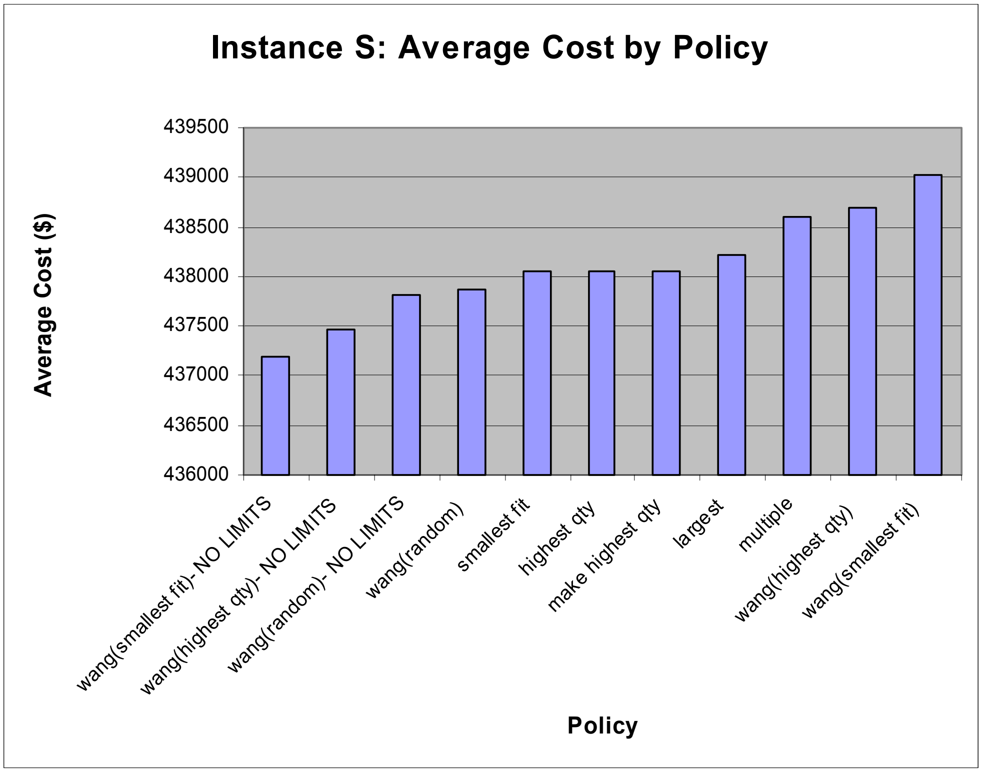

Figure 8: Instance S: Average Daily Cost Over All Replications by Policy

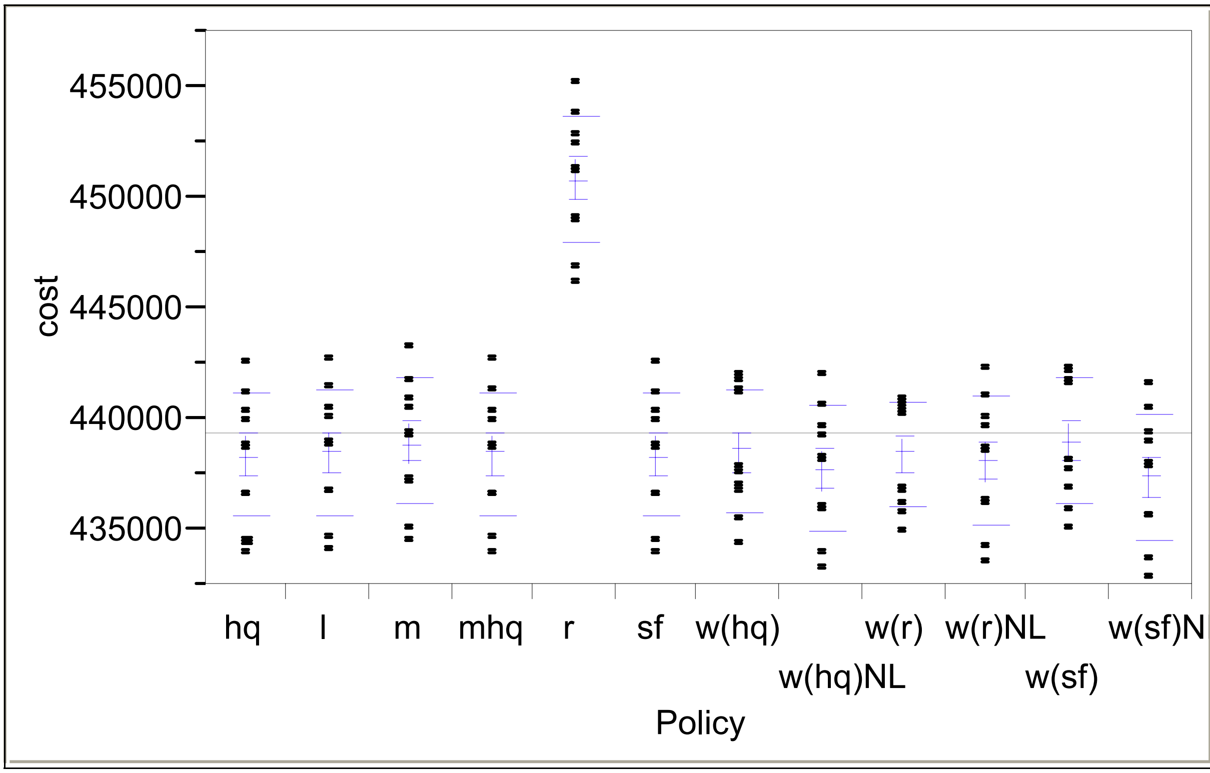

Figure 9: Instance S: 95% Confidence Intervals for Mean Cost by Policy

Table 1: Instance S: 95% Confidence Intervals for Mean Cost by Policy

	Lower 95%	Mean	Upper 95%
Random	448698	450804	452910
Wang (smallest fit)	436980	438944	440908
Multiple	436837	438879	440921
Wang (random)	436627	438343	440059
Wang (highest qty)	436499	438512	440525
Largest	436435	438471	440507
Make Highest Qty	436322	438365	440407
Highest Qty	436283	438323	440362
Smallest Fit	436283	438323	440362
Wang (random) NO LIMITS	435988	438059	440129
Wang (highest qty) NO LIMITS	435665	437704	439743
Wang (smallest fit) NO LIMITS	435353	437403	439453

Table 2: Instance S: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)

Random
Highest Qty, Largest, Multiple, Make Highest Qty, Smallest Fit, Wang(highest qty), Wang(random), Wang(smallest fit), Wang(highest qty)- NO LIMITS, Wang(random)- NO LIMITS, Wang(smallest fit)- NO LIMITS

In order to summarize our findings based on these confidence intervals, we created Table 2. Table 2 shows groupings of significantly different policies. In this case, the "random" policy is the only policy that performs significantly different from the others.

Now that we have examined the performance of each policy with respect to cost, we need to also consider the performance of our other measures. First, consider the ordered weight by replication, Figure 10. Careful examination of this graph shows that the average ordered weight only varies about 0.5% between replications. This measure shows us the relative order amounts per replication. Then, one might hypothesize that our other measures, cost, inventory weight, and scrap weight, may also follow a similar pattern.

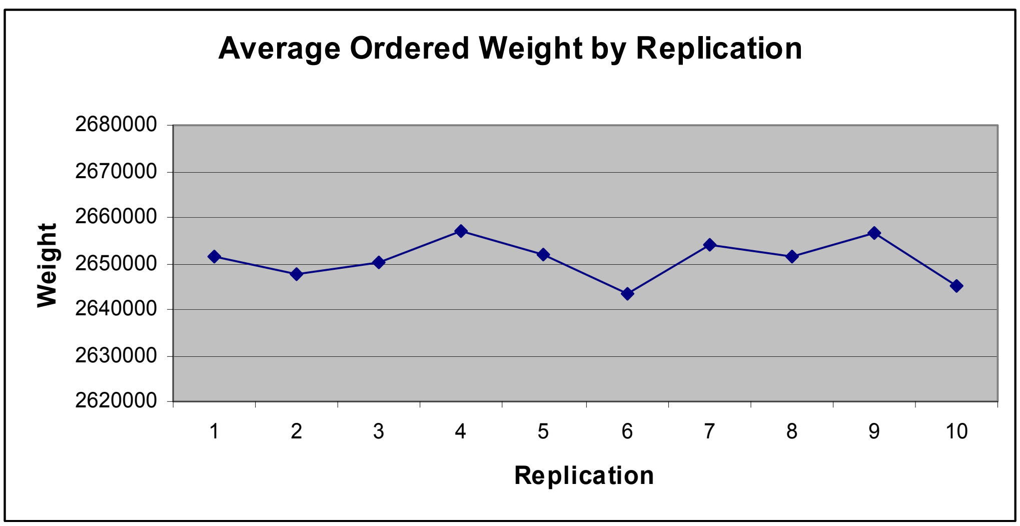

Figure 10: Instance S: Average Daily Ordered Weight by Replication

As expected, our average inventory weight (Figure 11) appears to vary linearly with the ordered weight per policy. Also, reconsider Figure 7, Cost by Policy, and notice that it too appears to vary linearly with respect to the ordered weight. This is an indication that our instance requires more steel and carries more inventory when there are more orders in the system.

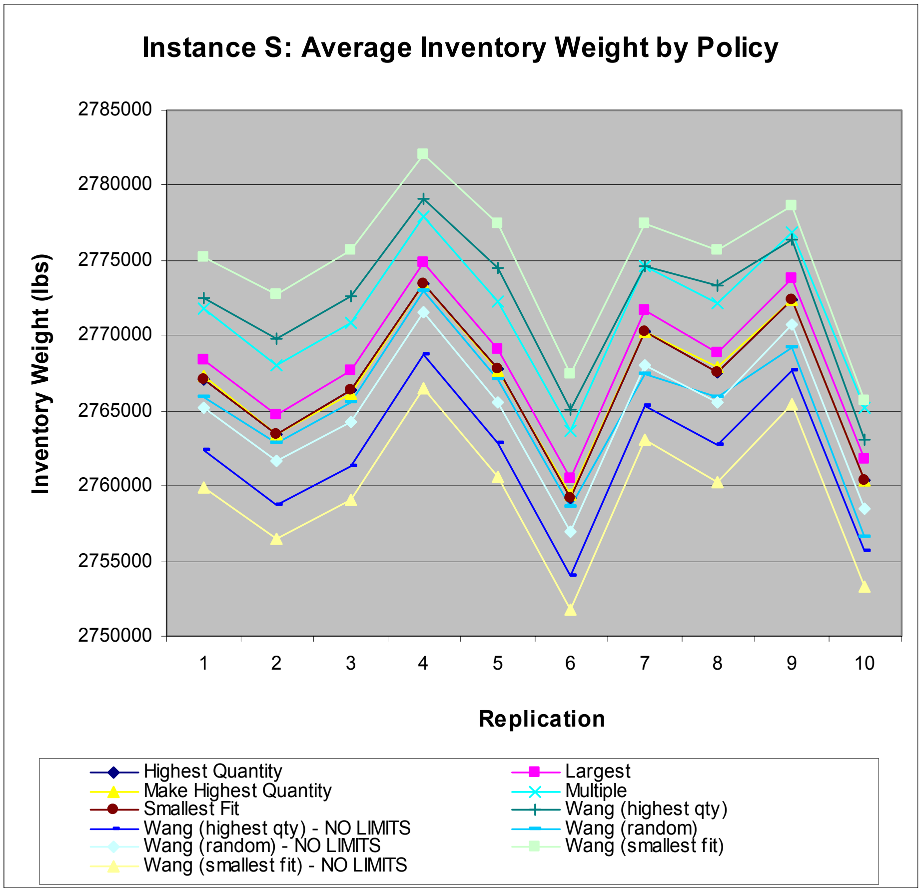

Figure 11: Instance S: Average Daily Inventory Weight by Policy

Similarly, Figure 12 displays the scrap weight by policy. It is difficult to see due to the scale of the graph, however, the scrap weight also seems to obey this linear relationship.

Another important note about all of these figures is the relative performance of each policy. Notice that in general, policies that have higher total costs also have higher inventory weights and higher scrap weights. Likewise, policies that have lower total costs also tend to have lower inventory weights and lower scrap weights.

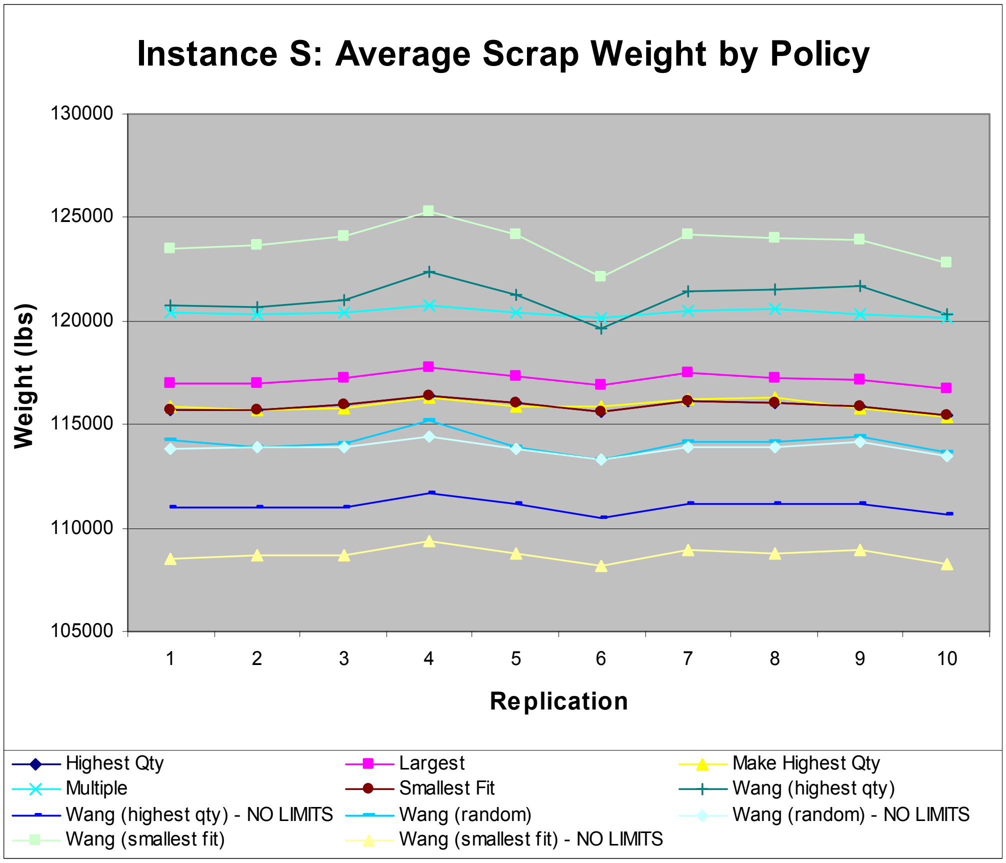

Figure 12: Instance S: Average Daily Scrap Weight by Policy

B. MULTIPLE DEMAND, MULTIPLE SUPPLY LOCATIONS

We extended our model to accommodate multiple supply and multiple demand locations. Once again, we investigated the policies described in Chapter 3, however, here we adjusted the policies to accommodate the multiple locations. For all heuristic policies expect for Wangs', the demand was filled from the nearest supply location whenever possible. In the few cases where that was not possible, demand was then fulfilled from the next closest facility. Other than this adjustment, the policies maintained their structures.

We developed a series of five similar instances (A-E) on the Euclidean Plane. Each instance has an identical set of supply locations (though the raw costs vary per instance) and a varying set of demand locations. However, the total demand rates per size remain constant across all instances. Figures 13-17 display the relative layouts of the instances. Here again, random is removed from the below graphs allowing us to better compare the more significant policies.

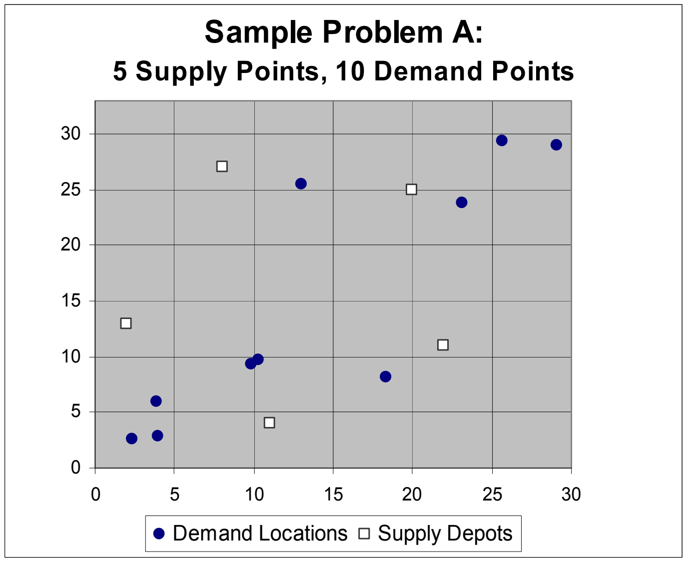

Figure 13: Layout of Sample Problem A

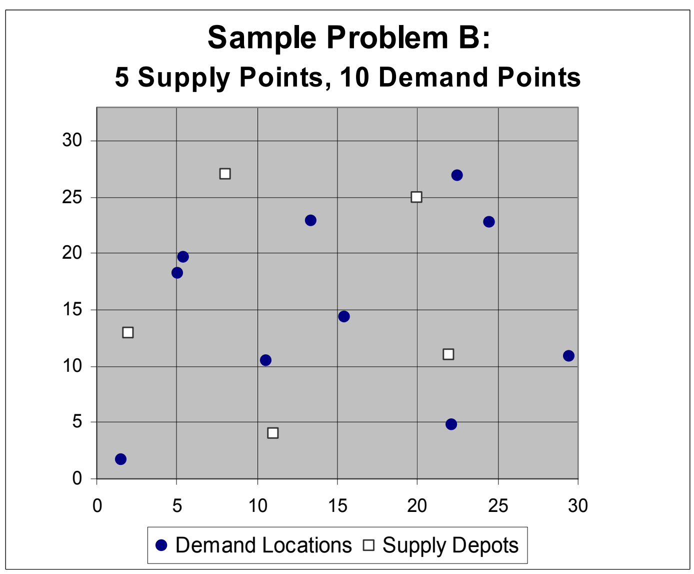

Figure 14: Layout of Sample Problem B

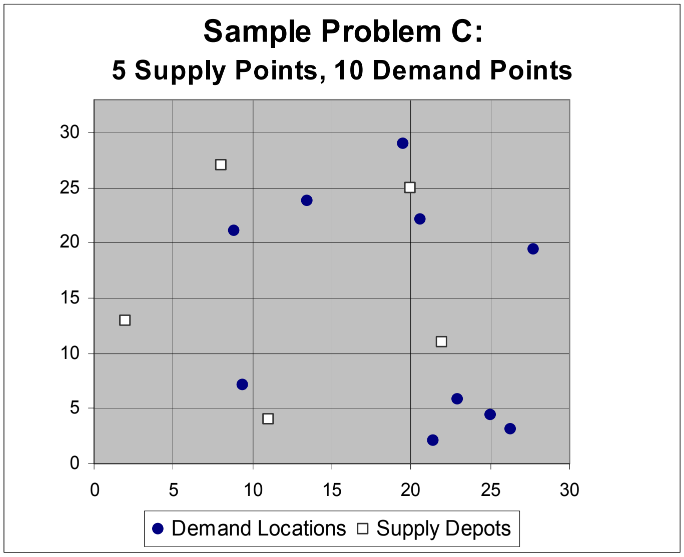

Figure 15: Layout of Sample Problem C

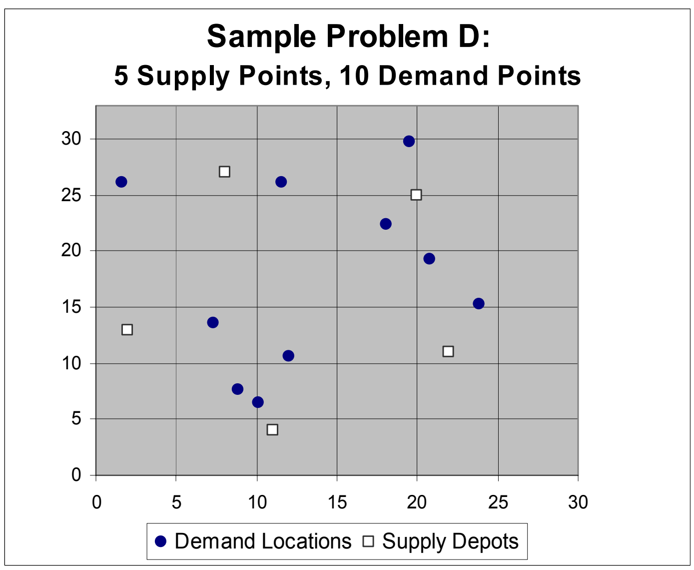

Figure 16: Layout of Sample Problem D

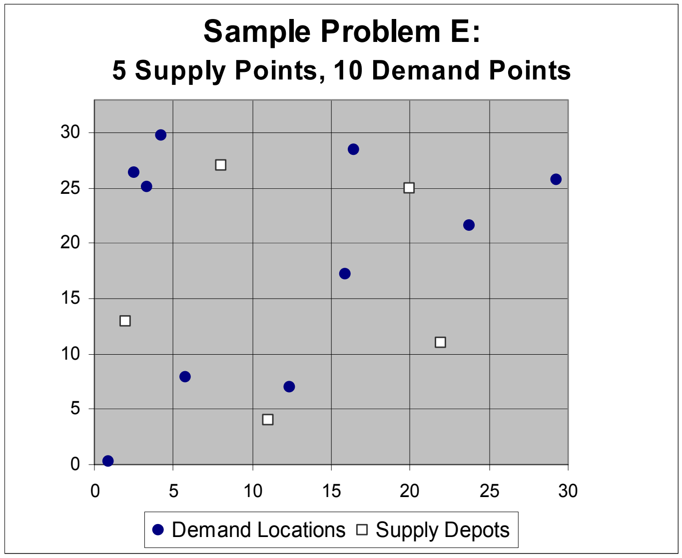

Figure 17: Layout of Sample Problem E

While instances A, B, C, D, and E are different (each is generated based on different random variables), their results are remarkably similar. Once again, if one considers the ordered weight for a instance and then compares the cost, inventory weight, and scrap weight, he will find predictable correlations between them all. Since all of the instances are so similar in this manner, we will only examine all performance measures for Instance A. Then, the costs and statistical performance of each policy in regards to costs will be analyzed for each policy.

1. Instance A

Figure 18 shows the average amount of weight ordered during each replication. Here too, the ordered weight varies minimally, 1.1% between replications at their maximum. This will again act as our baseline for comparing the performance of our other measures.

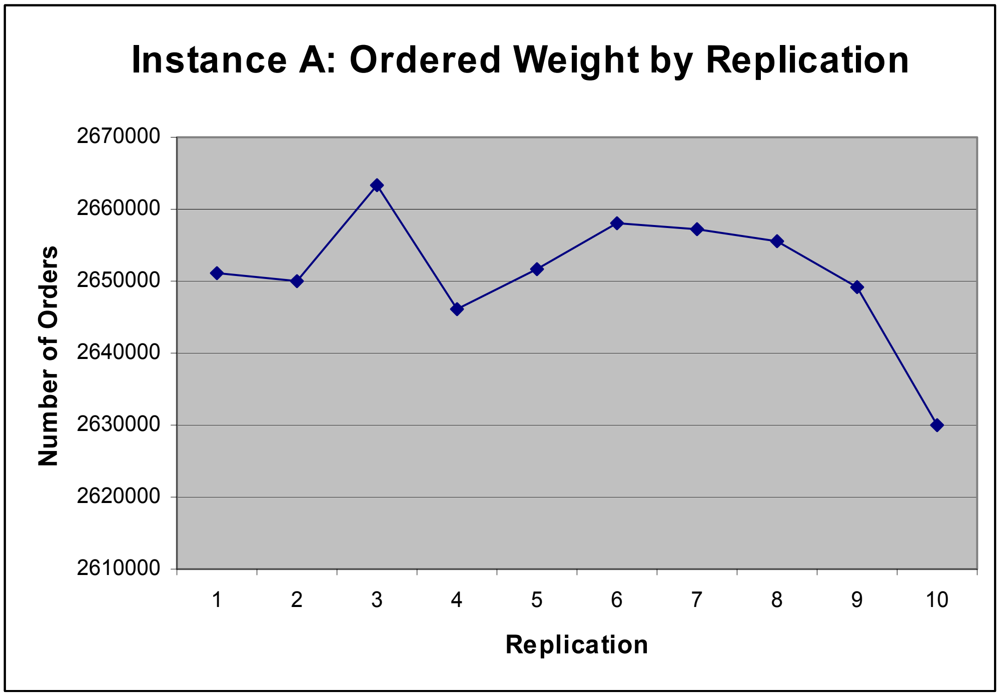

Figure 18: Instance A: Average Daily Ordered Weight

Inventory weight for each policy is displayed according to replication number in Figure 19. Although it may be difficult to discern the actual policies, there appear to be three groupings of policies. In Figure 20, the scrap weight for each policy, a similar pattern emerges. Also notice that all of the policies seem to perform somewhat linearly to the ordered weight. Finally, in Figure 21, the cost for each policy also follows these trends.

Note the relative positioning of each policy in each graph. That is, the policies that tend to have higher costs also tend to have higher inventory weights and higher scrap weights. Likewise, the policies with lower costs have smaller quantities in inventory and they also scrap less weight.

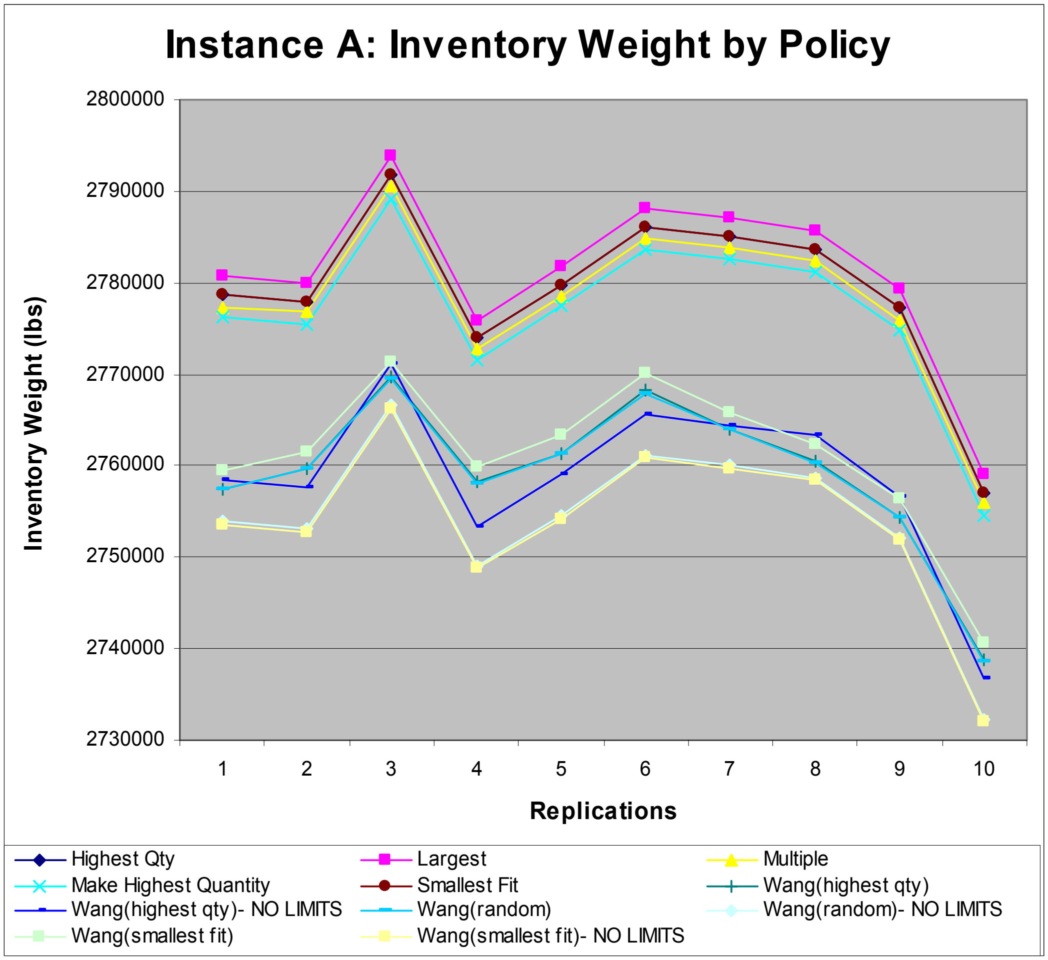

Figure 19: Instance A: Average Daily Inventory Weight by Policy

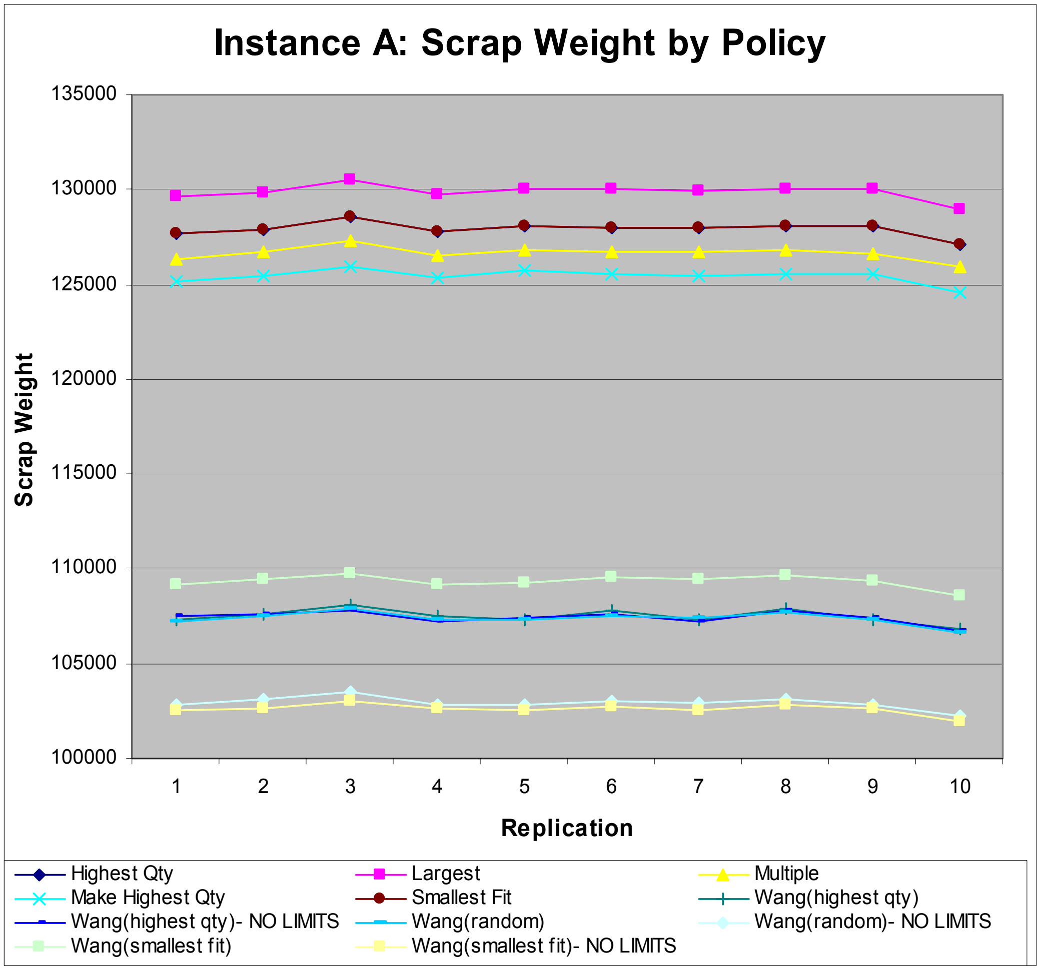

Figure 20: Instance A: Average Daily Scrap Weight by Policy

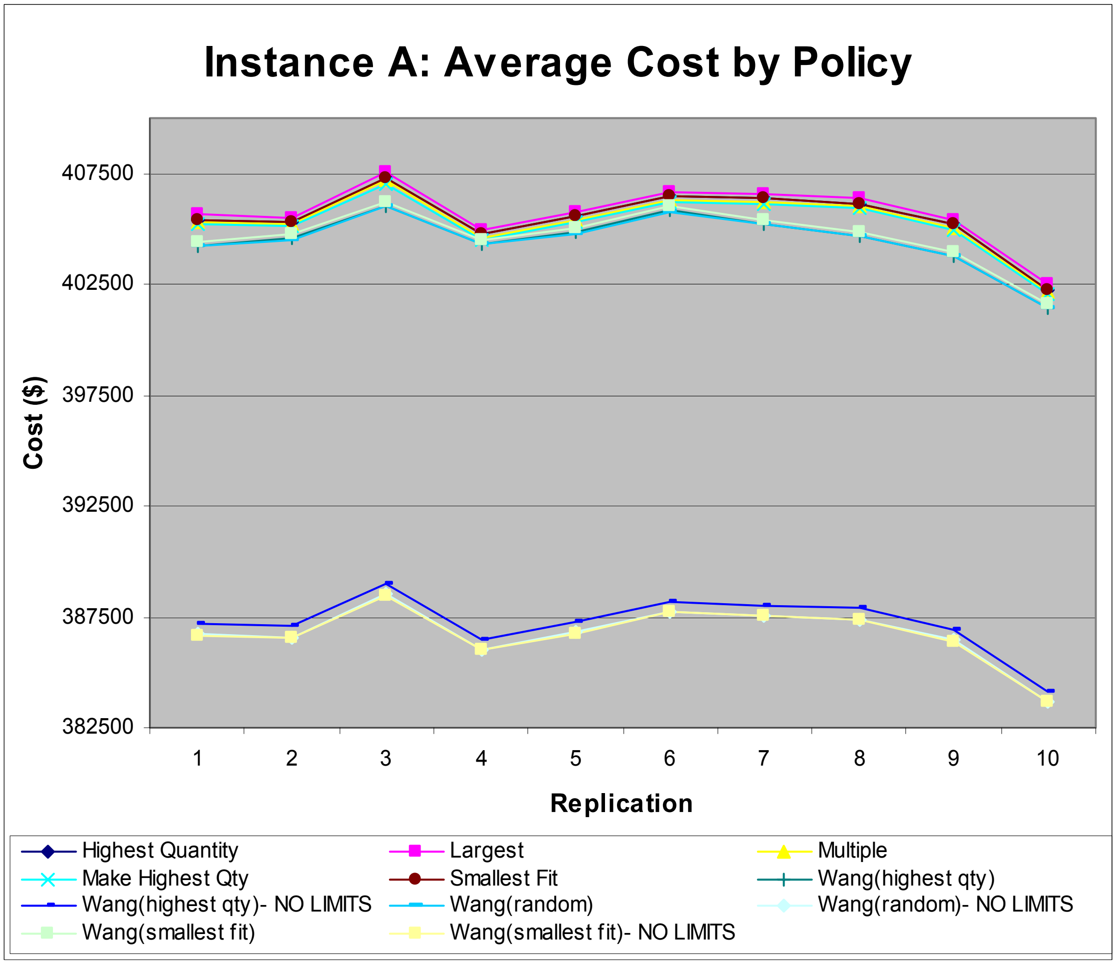

Figure 21: Instance A: Average Daily Cost by Policy

The costs for each policy shown in Figure 21 do appear to be obviously grouped according to performance, yet this figure alone is not enough to draw any conclusions. Figure 22 gives a summarized view of the same data. This figure allows one to see the relative performance of each policy, that is, one can discern which policies fall into which groupings.

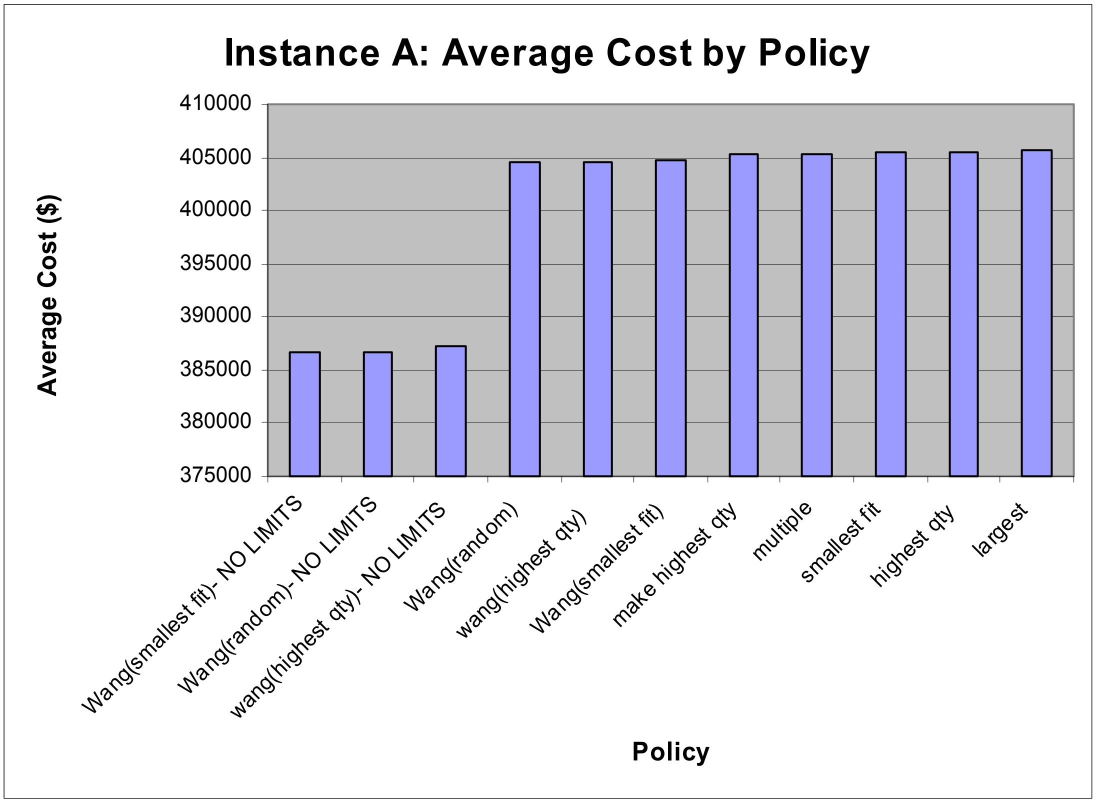

Figure 22: Instance A: Average Daily Cost by Policy over All Replications

Figure 23 and Table 3 show 95% confidence intervals for the mean cost for each policy. Policies are defined as significantly different if there 95% confidence intervals are completely independent. Thus, this information allows one to discern which policies are significantly different.

Our 95% confidence intervals are not inclusive of all replication means, and we are unable to conclusively give a reason for that at this time. We suspect that repeated replications and further statistical analysis would reveal that one of our replications is drawing on a bad seed value.

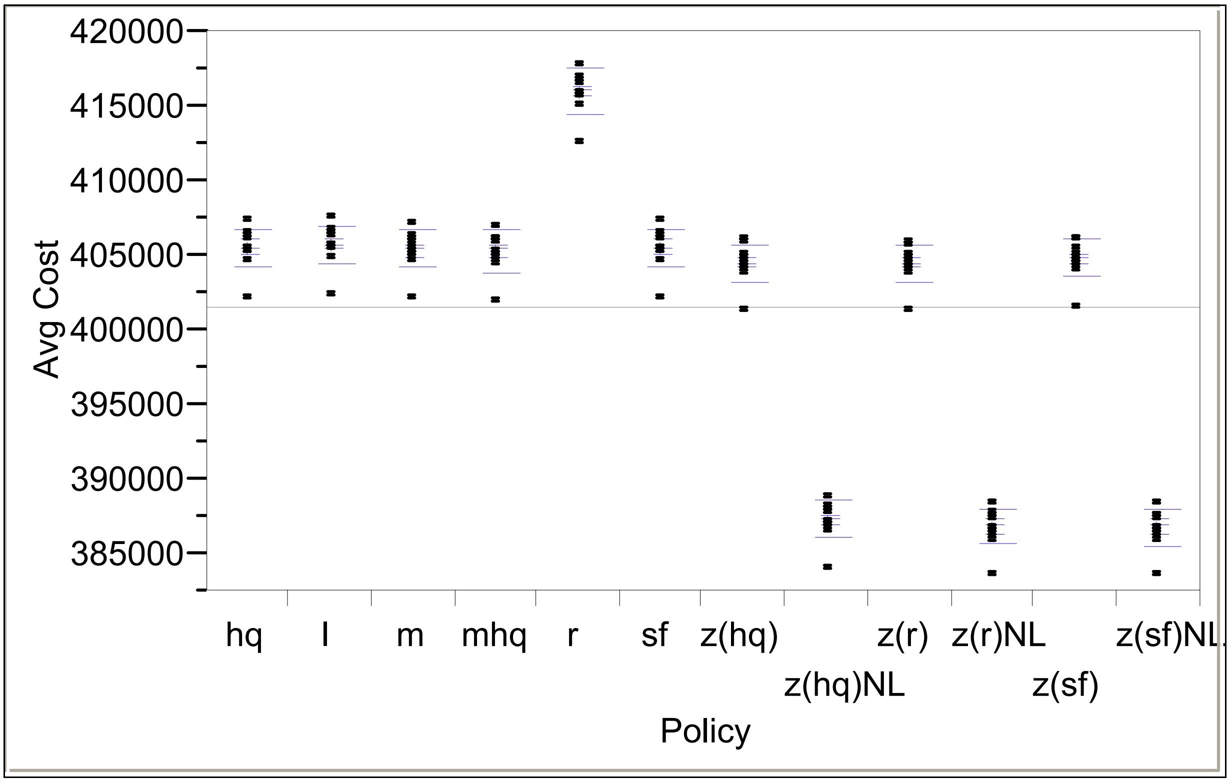

Figure 23: Instance A: 95% Confidence Intervals on the Average Daily Cost

Table 3: Instance A: 95% Confidence Intervals for the Mean Cost by Policy

	Lower 95%	Mean	Upper 95%
Random	414955	415938	416921
Largest	404729	405697	406665
Highest Qty	404521	405487	406453
Smallest Fit	404521	405487	406453
Multiple	404383	405347	406311
Make Highest Qty	404283	405247	406211
Wang (smallest fit)	403782	404702	405622
Wang (highest qty)	403566	404487	405407
Wang (random)	403555	404475	405395
Wang (highest qty) NO LIMITS	386282	387213	388144
Wang (random) NO LIMITS	385807	386738	387670
Wang (smallest fit) NO LIMITS	385775	386704	387633

Table 4 summarizes the groupings of significantly different policies for Instance A. The table is arranged such that the most expensive policy appears at the top of the chart and the least expensive policies appear at the bottom of the chart. Notice that there are only four significantly different levels of performance.

Table 4: Instance A: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)

Random
Highest Qty, Largest, Multiple, Make Highest Qty, Smallest Fit, Wang(highest qty), Wang(random), Wang(smallest fit)
Wang(highest qty)- NO LIMITS, Wang(random)- NO LIMITS, Wang(smallest fit)- NO LIMITS

2. Instance B

Similar analysis is performed for Instance B. First, the cost for each policy is examined in Figure 24. Notice that once again, four groupings emerge in the graph. These policies may also be viewed side-by-side in Figure 25.

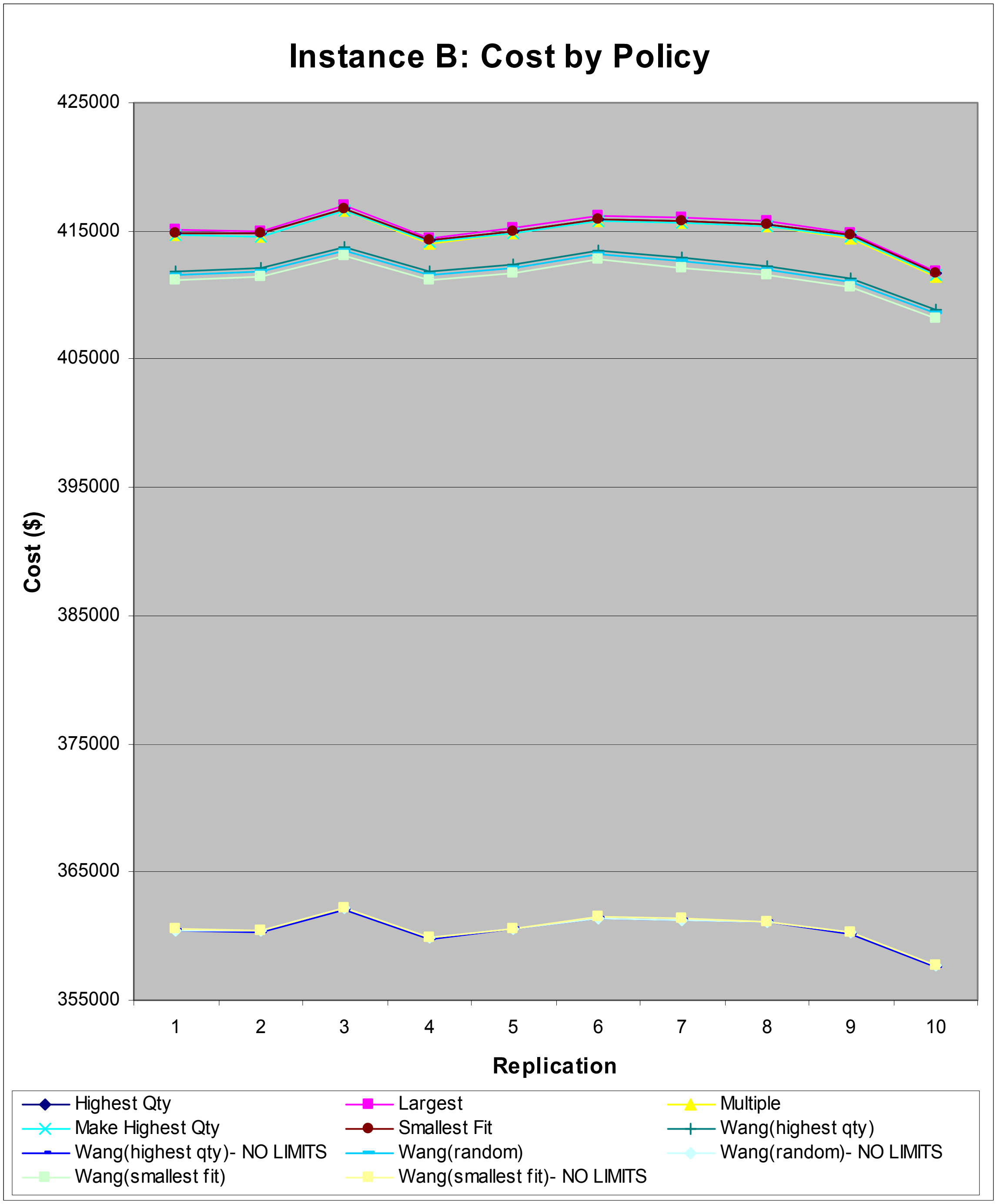

Figure 24: Instance B: Average Daily Cost for Each Policy

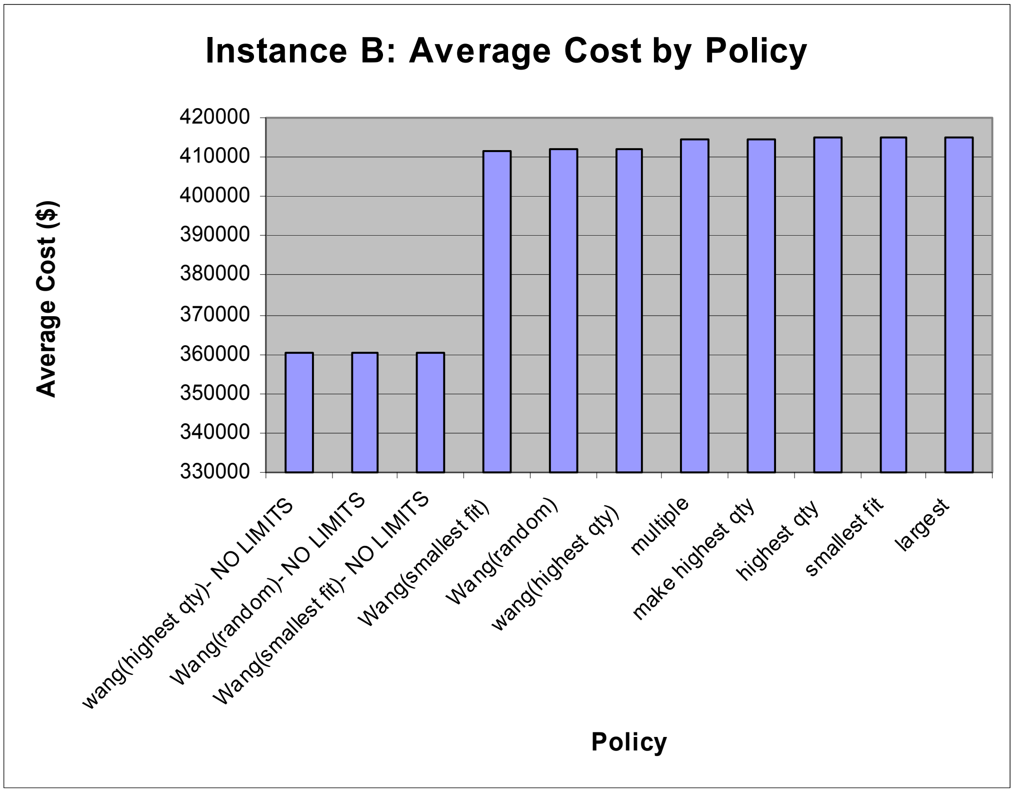

Figure 25: Instance B: Average Daily Cost by Policy over All Replications

Figure 26 and Table 5 show 95% confidence intervals for mean cost per policy. The confidence intervals are used to confirm which policies are statistically different. We have grouped like policies into each level of Table 6 and have ordered them from the highest cost grouping at the top to the lowest cost grouping at the bottom.

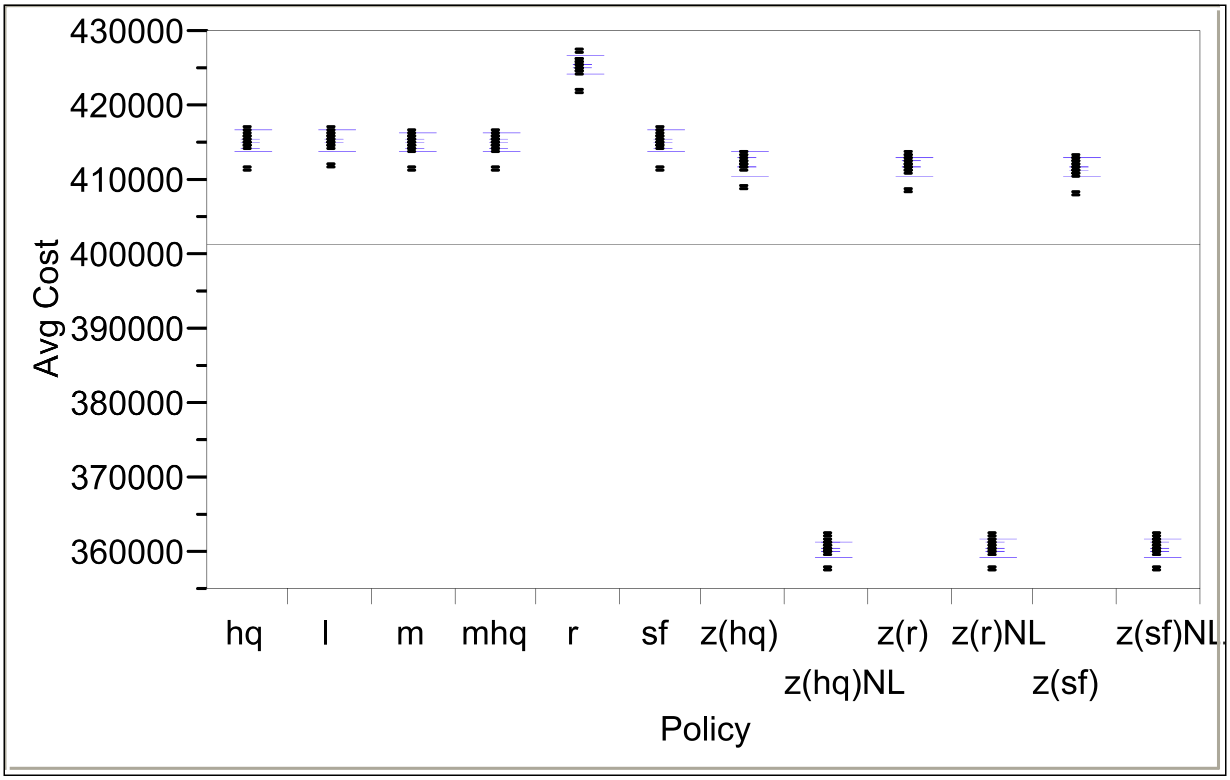

Figure 26: Instance B: 95% Confidence Intervals for Mean Daily Cost for Each Policy

Table 5: Instance B: 95% Confidence Intervals for Mean Daily Cost

	Lower 95%	Mean	Upper 95%
Random	424214	425209	426204
Largest	414132	415110	416087
Highest Qty	413944	414922	415899
Smallest Fit	413944	414922	415899
Make Highest Qty	413798	414774	415750
Multiple	413754	414734	415713
Wang (highest qty)	411088	412070	413052
Wang (random)	410824	411805	412786
Wang (smallest fit)	410417	411396	412375
Wang (smallest fit) NO LIMITS	359684	360550	361416
Wang (random) NO LIMITS	359647	360513	361378
Wang (highest qty) NO LIMITS	359592	360457	361322

"Random" is significantly different from all others. Wang's policies with limiting are not significantly different from one another, yet, the three of them are all significantly different from all of the other policies. Similarly, Wang's' policies without limiting are not significantly different from one another, yet, the three of them are all significantly different from all of the other policies. The remaining policies (highest quantity, largest, multiple, make highest quantity, and smallest fit) are not significantly different from one another. These conclusions are summarized in Table 6.

.

Table 6: Instance B: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)

Random
Highest Qty, Largest, Multiple, Make Highest Qty, Smallest Fit
Wang(highest qty), Wang(random), Wang(smallest fit)
Wang(highest qty)- NO LIMITS, Wang(random)- NO LIMITS, Wang(smallest fit)- NO LIMITS

3. Instance C

The costs of each policy are summarized in Figure 27. There appear to be three groupings in this graph, though one of the groups is considerably more variant than the others. Therefore, it is beneficial to view a summary by policy as in Figure 28. In this graph, four groups appear to emerge, though we are unable to draw that conclusion with any certainty.

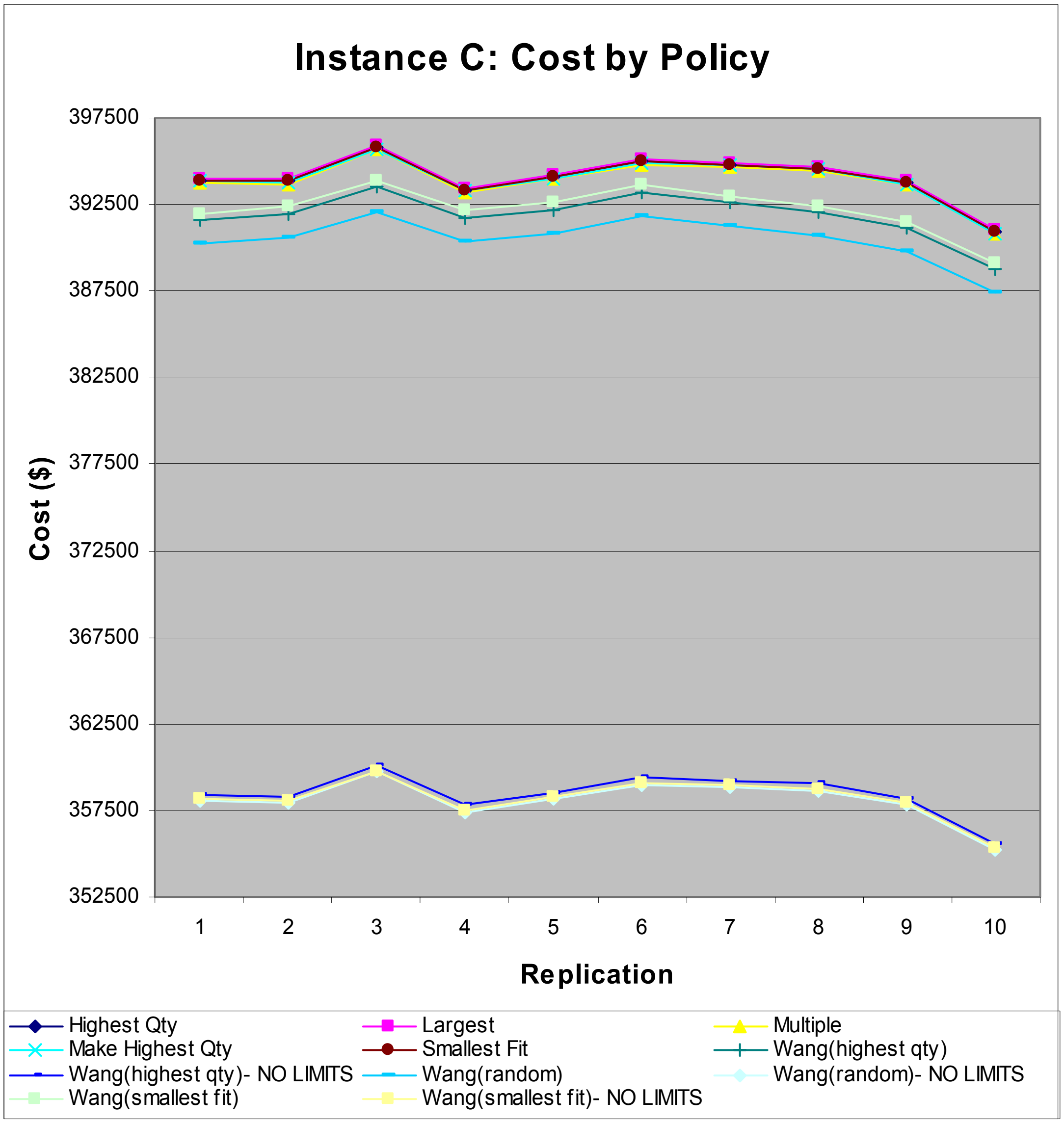

Figure 27: Instance C: Average Daily Cost by Policy

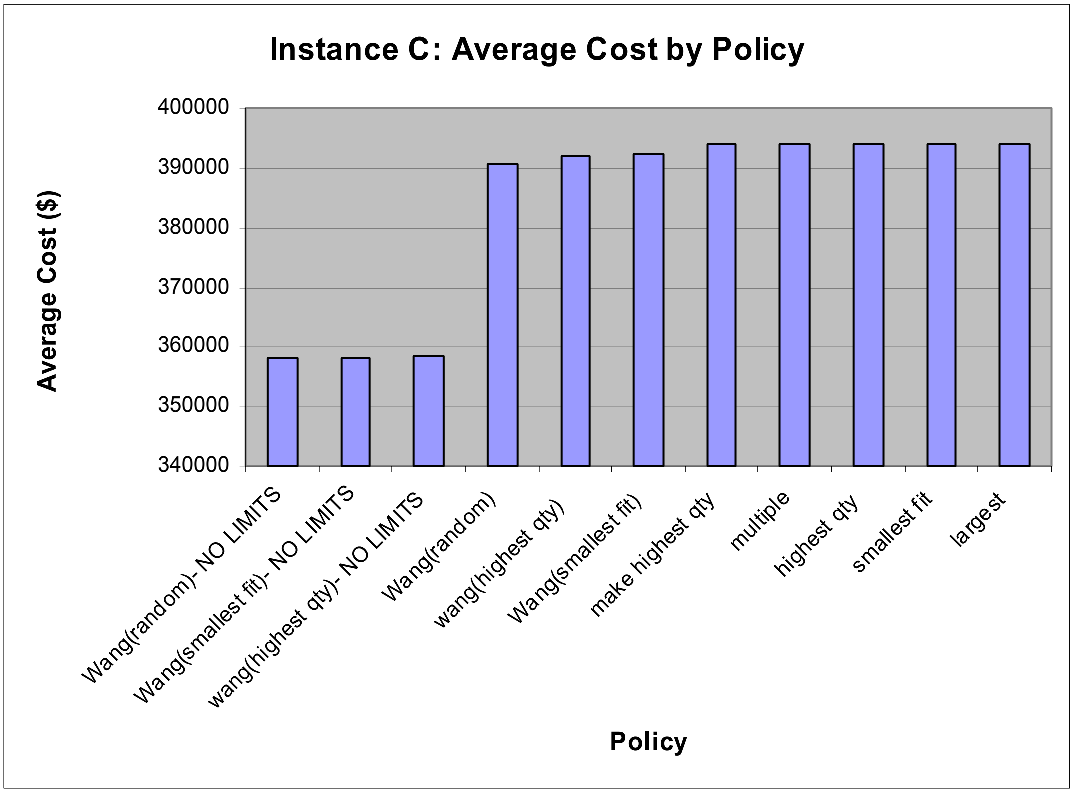

Figure 28: Instance C: Average Daily Cost by Policy over All Replications

Figure 29 and Table 7 show 95% confidence intervals for mean cost for each policy. This figure and this table allow us to statistically confirm five groupings: random; highest quantity, large, multiple, make highest quantity, and smallest fit; Wang (highest quantity) and Wang (smallest fit); Wang (random); and Wang (highest quantity) - NO LIMITS, Wang (smallest fit) - NO LIMITS, and Wang (random) - NO LIMITS. These groupings are summarized in Table 8.

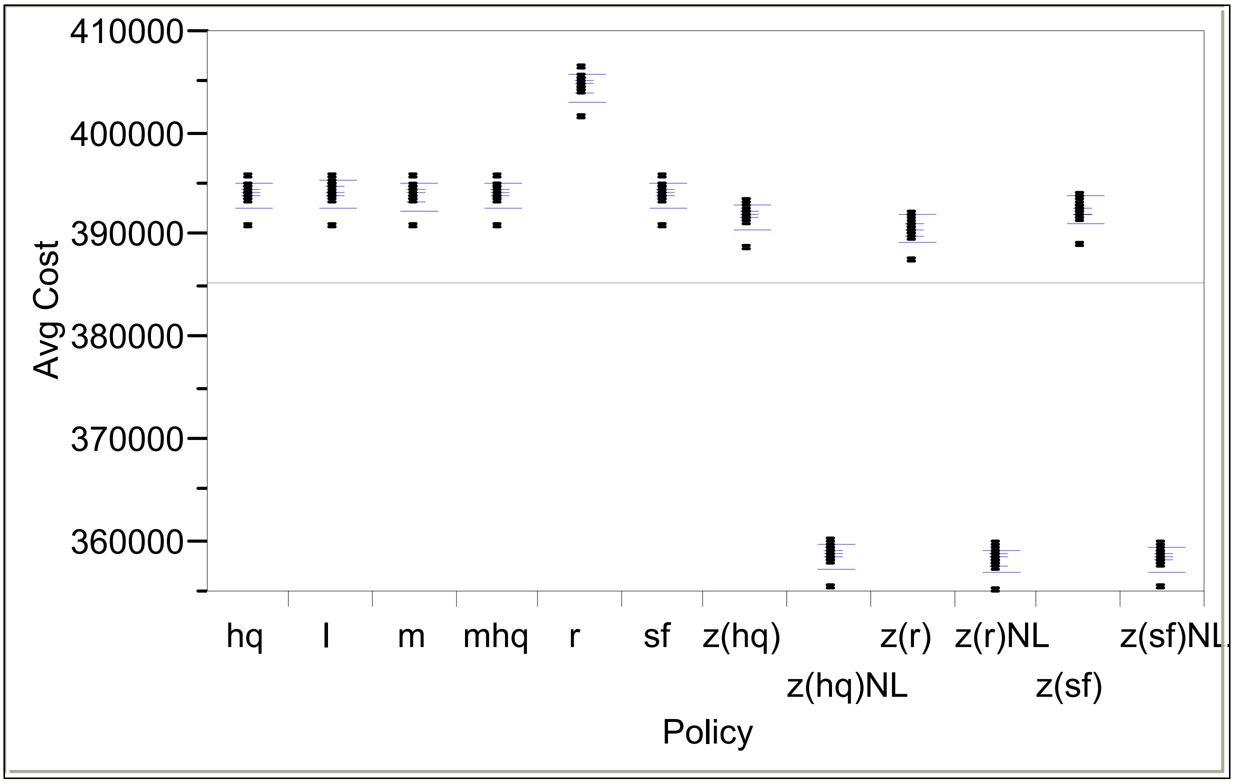

Figure 29: Instance C: 95% Confidence Intervals for Mean Cost for Each Policy

Table 7: Instance C: 95% Confidence Intervals for Mean Cost for Each Policy

	Lower 95%	Mean	Upper 95%
Random	403645	404587	405528
Largest	393166	394103	395040
Highest Qty	393049	393986	394923
Smallest Fit	393049	393987	394924
Make Highest Qty	392995	393931	394866
Multiple	392939	393871	394804
Wang (smallest fit)	391321	392266	393211
Wang (highest qty)	390924	391871	392817
Wang (random)	389584	390503	391422
Wang (highest qty) NO LIMITS	357601	358463	359326
Wang (smallest fit) NO LIMITS	357306	358163	359020
Wang (random) NO LIMITS	357204	358067	358930

Table 8: Instance C: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)

Random
Highest Qty, Largest, Multiple, Make Highest Qty, Smallest Fit
Wang(highest qty), Wang(smallest fit)
Wang(random)
Wang(highest qty)- NO LIMITS, Wang(random)- NO LIMITS, Wang(smallest fit)- NO LIMITS

4. Instance D

Figure 30 shows the cost of fulfilling all orders for each policy. There appear to be five different groupings of cost for this instance. In Figure 31, we see the averages across all replications of the costs using each policy. Here it is difficult to discern whether there are four or five distinct groups.

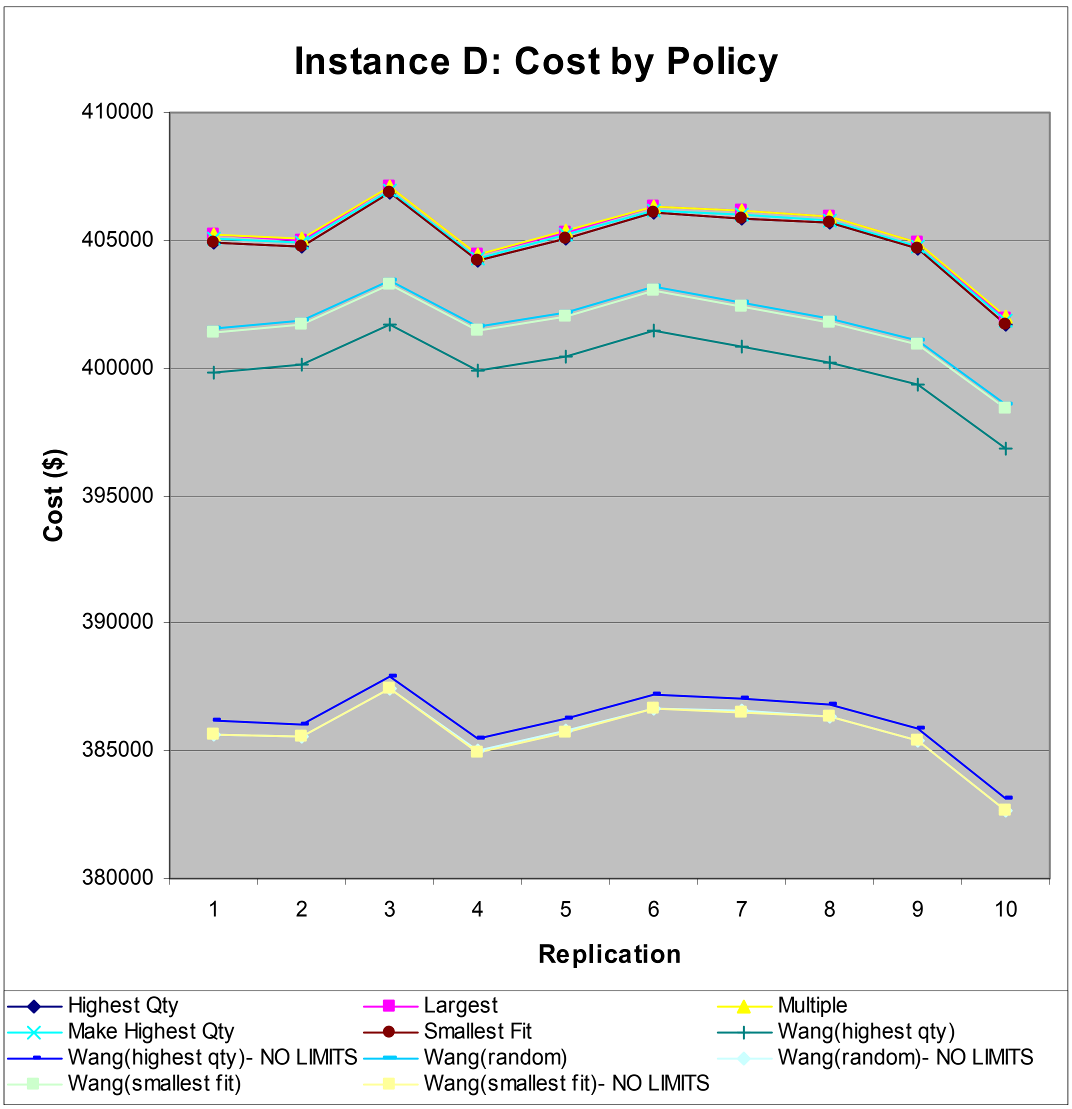

Figure 30: Instance D: Average Daily Cost by Policy

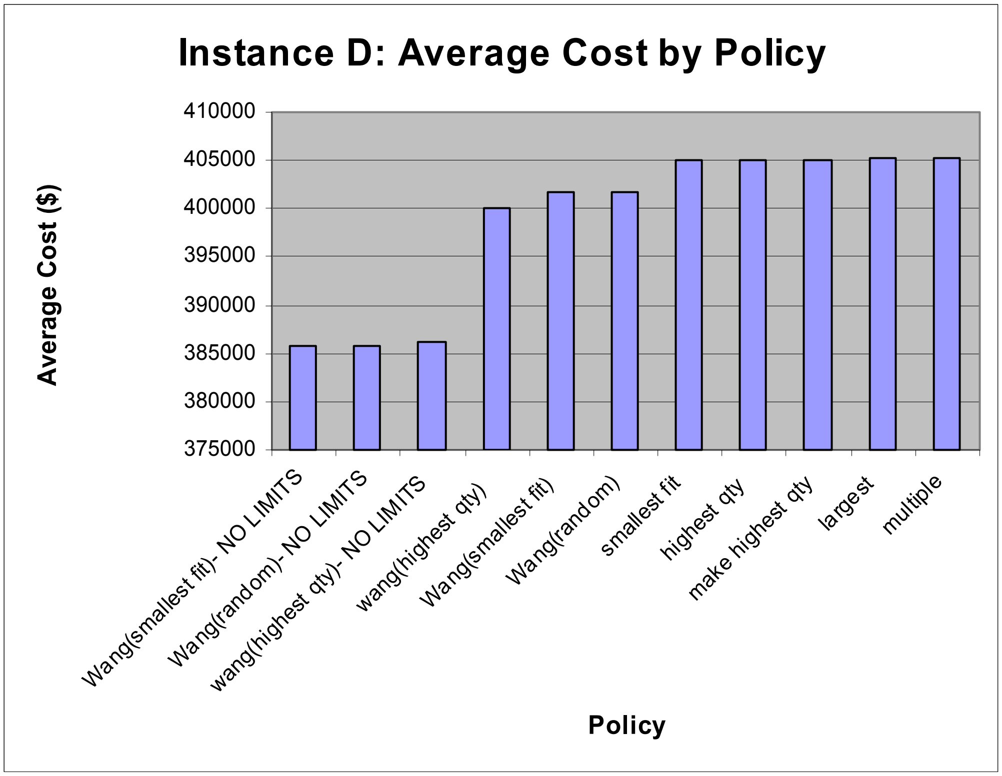

Figure 31: Instance D: Average Daily Cost by Policy over All Replications

Ninety-five percent confidence intervals for the mean cost per replication are shown in Figure 32 and in Table 9. These confidence intervals allow us to discern five individual groupings of distinct policies. These groupings are summarized in Table 10.

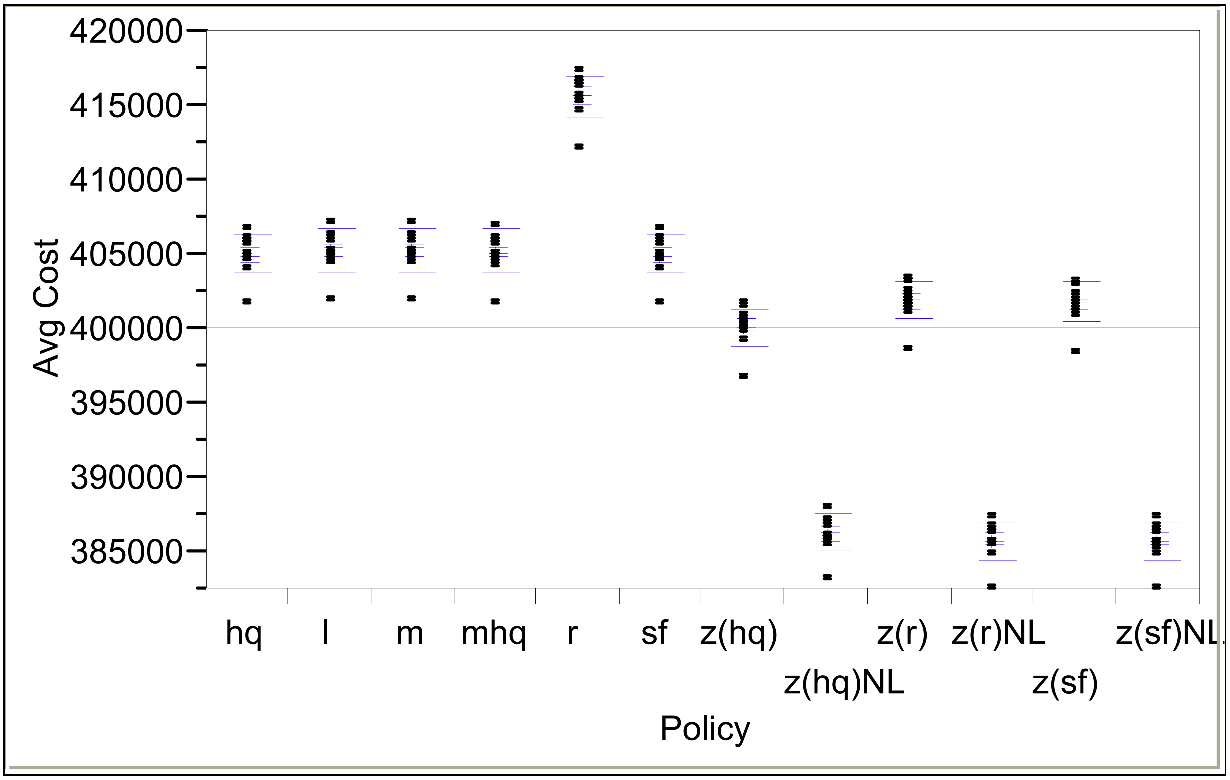

Figure 32: Instance D: 95% Confidence Intervals of Mean Cost for Each Policy

Table 9: Instance D: 95% Confidence Intervals for Mean Cost

	Lower 95%	Mean	Upper 95%
Random	414579	415594	416609
Multiple	404255	405250	406245
Largest	404223	405222	406222
Make Highest Qty	404101	405095	406089
Highest Qty	403971	404968	405965
Smallest Fit	403971	404968	405965
Wang (random)	400827	401800	402773
Wang (smallest fit)	400660	401635	402611
Wang (highest qty)	399101	400074	401047
Wang (highest qty) NO LIMITS	385259	386184	387110
Wang (random) NO LIMITS	384786	385713	386639
Wang (smallest fit) NO LIMITS	384760	385688	386616

Table 10: Instance D: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)

Random
Highest Qty, Largest, Multiple, Make Highest Qty, Smallest Fit
Wang(random), Wang(smallest fit)
Wang(highest qty)
Wang(highest qty)- NO LIMITS, Wang(random)- NO LIMITS, Wang(smallest fit)- NO LIMITS

5. Instance E

The cost of each policy is displayed in Figure 33. There appear to be three groupings of policies based on this graph. In order to more clearly view how each policy performs, costs were averaged over all replications and are summarized in Figure 34. This instance is interesting, because Wang's policies all appear to perform equally well, even those that are subject to incoming inventory limits.

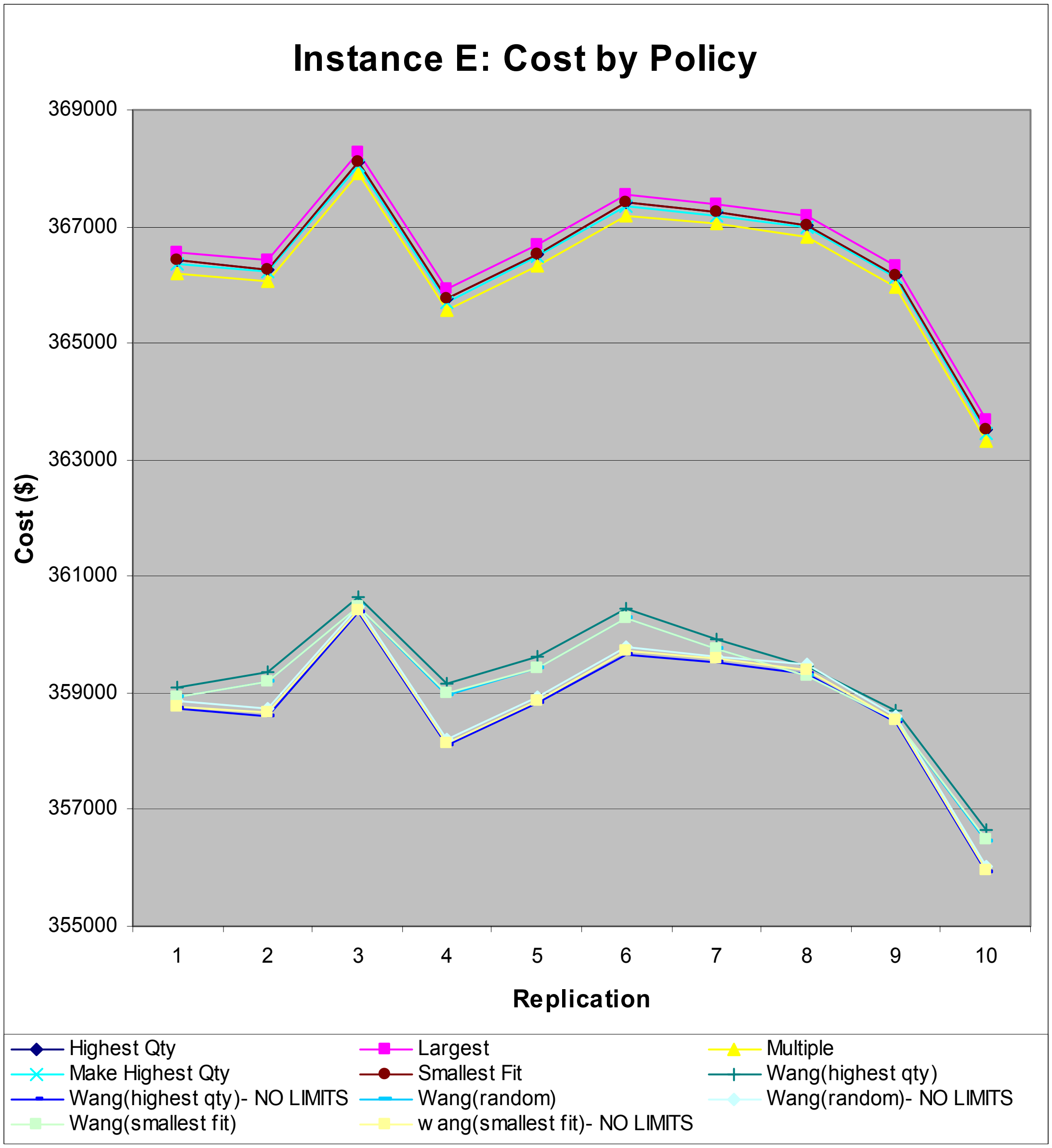

Figure 33: Instance E: Average Daily Cost by Policy

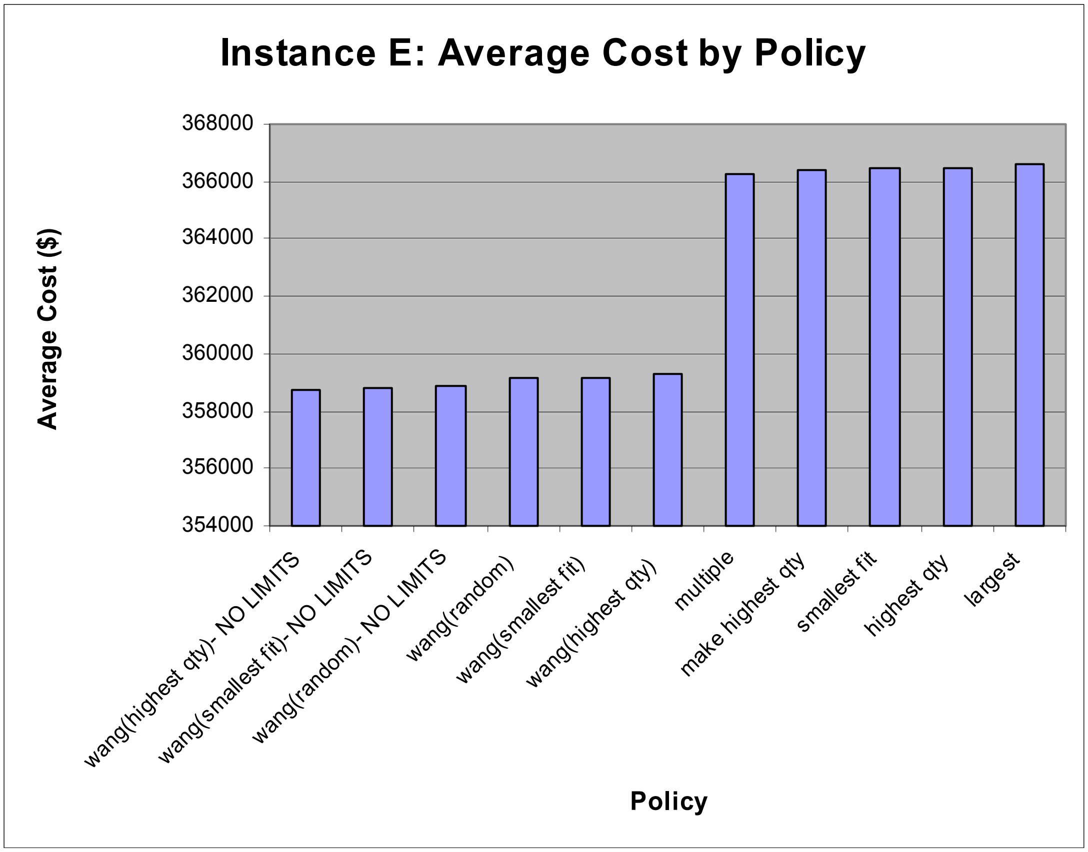

Figure 34: Instance E: Average Daily Cost by Policy Over All Replications

Figure 35 and Table 11 show 95% confidence intervals for the mean cost of each policy. Here we are able to statistically confirm that there is not a significant difference in performance for Wang's policies, regardless of tie-breaking rule or incoming inventory restrictions. Table 12 summarizes the groupings of significantly different policies.

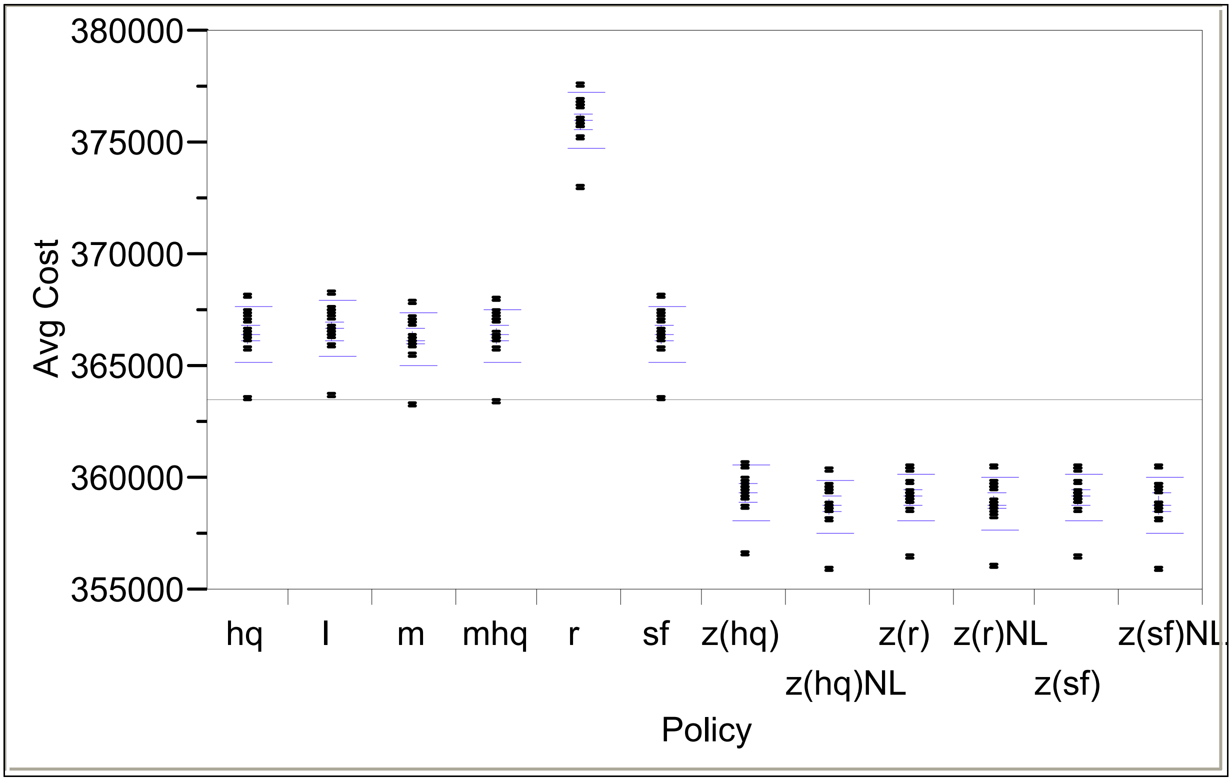

Figure 35: **Instance E: 95% Confidence Intervals for Mean Cost for Each Policy**

Table 11: Instance E: 95% Confidence Intervals for Mean Cost

	Lower 95%	Mean	Upper 95%
Random	375071	375966	376861
Largest	365714	366601	367487
Highest Qty	365555	366442	367329
Smallest Fit	365555	366442	367329
Make Highest Qty	365506	366394	367281
Multiple	365358	366242	367126
Wang (highest qty)	358510	359306	360102
Wang (smallest fit)	358344	359140	359935
Wang (random)	358337	359136	359934
Wang (random) NO LIMITS	358012	358872	359733
Wang (smallest fit) NO LIMITS	357946	358807	359667
Wang (highest qty) NO LIMITS	357899	358758	359617

Table 12: Instance E: Groupings of Significantly Different Policies, Highest Cost (top) to Lowest Cost (bottom)

Random
Highest Qty, Largest, Multiple, Make Highest Qty, Smallest Fit
Wang(highest qty), Wang(random), Wang(smallest fit), Wang(highest qty)- NO LIMITS, Wang(random)- NO LIMITS, Wang(smallest fit)- NO LIMITS

C. SUMMARY

Summary charts of Daily Costs, Inventory Weight and Scrap Weight are shown in Tables 13, 15, and 16, respectively. In each case, the smallest performance measure was set to 1.00 for each model and then each other measure was divided by this smallest measure to examine the relative performance of each policy. Table 14 provides a graphical summary of the Daily Cost information provided in Table 13.

Table 13: Relative Performance of Daily Cost Across all Instances

	Random	Highest Qty	Largest	Multiple	Make Highest Qty	Smallest Fit	Wang (smallest fit)	Wang (random)	Wang (highest qty)	Wang (highest qty)- NO LIMITS	Wang (random)- NO LIMITS	Wang (smallest fit)- NO LIMITS
S	1.03	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
A	1.08	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.00	1.00	1.00
B	1.18	1.15	1.15	1.15	1.15	1.15	1.14	1.14	1.14	1.00	1.00	1.00
C	1.13	1.10	1.10	1.10	1.10	1.10	1.10	1.09	1.09	1.00	1.00	1.00
D	1.08	1.05	1.05	1.05	1.05	1.05	1.04	1.04	1.04	1.00	1.00	1.00
E	1.05	1.02	1.02	1.02	1.02	1.02	1.00	1.00	1.00	1.00	1.00	1.00

Table 14: Graphical View of Daily Cost by Instance

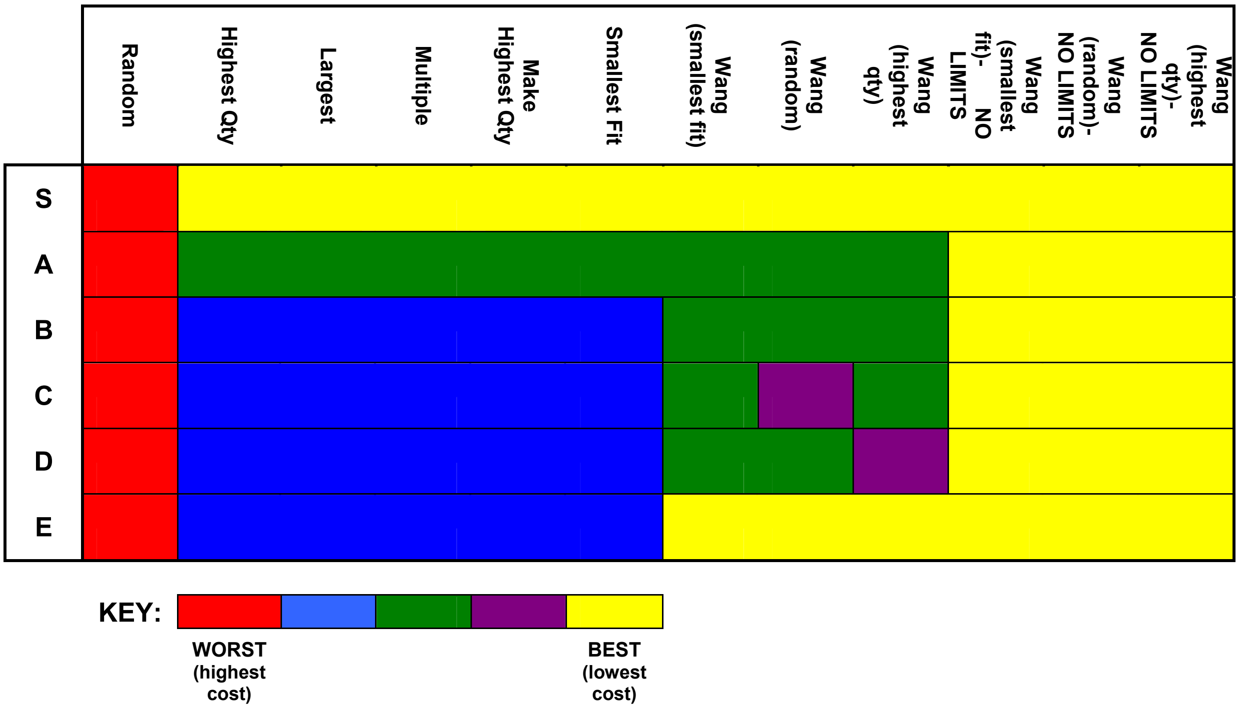

Table 15: Relative Performance of Daily Scrap Weight Across all Instances

	Random	Highest Qty	Largest	Multiple	Make Highest Qty	Smallest Fit	Wang (smallest fit)	Wang (random)	Wang (highest qty)	Wang (highest qty)-NO LIMITS	Wang (random)-NO LIMITS	Wang (smallest fit)-NO LIMITS
S	2.02	1.07	1.08	1.11	1.07	1.07	1.14	1.05	1.11	1.00	1.05	1.02
A	2.16	1.25	1.27	1.23	1.22	1.25	1.07	1.05	1.05	1.00	1.00	1.05
B	2.17	1.31	1.33	1.29	1.30	1.31	1.00	1.03	1.05	1.09	1.09	1.08
C	2.13	1.18	1.19	1.17	1.17	1.18	1.09	1.06	1.06	1.01	1.00	1.04
D	2.15	1.19	1.21	1.21	1.20	1.19	1.00	1.01	1.00	1.03	1.03	1.08
E	2.25	1.29	1.31	1.27	1.28	1.29	1.05	1.05	1.06	1.01	1.01	1.00

Table 16: Relative Performance of Daily Inventory Weight Across all Instances

	Random	Highest Qty	Largest	Multiple	Make Highest Qty	Smallest Fit	Wang (smallest fit)	Wang (random)	Wang (highest qty)	Wang (highest qty) - NO LIMITS	Wang (random) - NO LIMITS	Wang (smallest fit)- NO LIMITS
S	1.04	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00
A	1.04	1.01	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00
B	1.04	1.01	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00
C	1.04	1.01	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00
D	1.04	1.01	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00
E	1.05	1.01	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00

CHAPTER V

CONCLUSIONS AND EXTENSIONS

We were successful in simulating the process by which a steel firm receives orders and then decides how to handle them. We were also able to develop several other interesting, yet intuitive policies for comparison. Our model outputs various performance measures which enable us to compare and contrast the performance of each policy.

Overall, the results show that any basic policy provides a statistical advantage over randomly selecting a piece to cut from. Moreover, for a single supply, single demand instance, Wang's policy gives no statistical advantage over any other basic policy. However, when the system is expanded to include multiple supply and multiple demand points, Wang's policy shows a significant advantage over the other policies tested.

We were unable to show that using the consumption rates from Wang's model could improve his policy's performance. In fact, in many cases, Wang's policy performed significantly worse when incoming inventory restrictions (limits) were implemented.

Although we applied our model to the specific case of Straightline, we believe that it is very generalizable to other industries such as: fiber optics, paper, and plastics. This application may require small adjustments to the model to consider the nuances of each industry, but fundamentally they are very similar to the steel industry.

Our work on this project could be expanded by more broadly considering incoming inventory restrictions. Also, we could reconsider how to use the inventory consumption rates from Wang's output to improve his policy's performance, rather than hindering it.

There is also an opportunity to expand the model to include more detail, such as varied costs of cutting based on the size of the piece cut, an inventory cost based on the amount of time that a piece is held in inventory, and most importantly, expanding the model to consider two-dimensional cutting.

BIBLIOGRAPHY

1. Adelman, D., “Remnant Inventory Systems,” Ph.D. Thesis, Georgia Institute of Technology, 1997.

2. Adelman, D. and G. L. Nemhauser, “Price-Directed Control of Remnant Inventory Systems,” *Operations Research*, Vol. 47, pp. 889-898, 1999.

3. Banks, J., et al., Discrete-Event System Simulation, Upper Saddle River, New Jersey: Prentice Hall, 2001.

4. Dyckhoff, H., “A Typology of Cutting and Packing Problems,” *European Journal of Operational Research*, Vol. 44, pp. 145-159, 1990.

5. “Foreign Specialty Steel Extends Reach Into U S Market,” 8 July 2003, Specialty Steel Industry of North America, November 2003. <http://www.ssina.com/SSINA2002/SSINAsite/news/releases/pdf_releases/7_17_03imports.pdf>.

6. Goulimis, C., “Optimal Solutions for the Cutting Stock Problem,” *European Journal of Operational Research*, Vol. 44, pp. 197-208, 1990.

7. Haessler, R. W., “Controlling Cutting Pattern Changes in One-Dimensional Trim Problems,” *Operations Research*, Vol. 23, pp. 483-493, 1975.

8. Haessler, R. W. and P. E. Sweeney, “Cutting Stock Problems and Solution Procedures,” *European Journal of Operational Research*, Vol. 54, pp. 141-150, 1991.

9. Krichagina, E. V., R. Rubio, M. I. Taksar, and L. M. Wein, “A Dynamic, Stochastic Stock-Cutting Problem,” *Operations Research*, Vol. 46, pp. 690-701, 1998.

10. Matsumoto, M. and T. Nishimura, “Mersenne Twister Home Page,” Updated 18 Feb 2002, <http://www.math.keio.ac.jp/~matumoto/emt.html>.

11. “Monthly Statistics: September 2003,” 21 October 2003, *International Iron and Steel Institute*, November 2003. <http://www.worldsteel.org>.

12. “NIST Industrial Impact,” March 1994, *National Institute of Standards and Technology*, November 2003. <http://www.nist.gov/public_affairs/factsheet/steel.htm>.

13. Nonato, M. and M. G. Scutella, “The One-Dimensional Cutting Stock Problem: A Linear Programming Algorithm Based on Hyperflows,” *Proceedings of Algorithms and Experiments*, Edited by R. Battiti and A. A. Bertossi, pp. 44-50, 1998.

14. “Straightline,” 2003, *Straight Source a Division of United States Steel Corporation*, October 2003. <www.straightline.com>.

15. Wang, Z., A. Schaefer, and J. Rajgopal, “Optimal Operation of Supply Chains for Remnant Inventory Systems,” Working paper, 2003.