ABSTRACT

BERRINGS, LAUREN M. State clustering in Markov Decision Processes with An Application in Information Sharing.
(Under the direction of Dr. Russell E. King and Dr. Thom J. Hodgson)

This research examines state clustering in Markov Decision processes, specifically addressing the problem referred to as Markov Decision process with restricted observations. The general problem is a special case of a Partially Observable Markov Decision process where the state space is partitioned into mutually exclusive sets representing the observable portion of the process. The goal is to find an optimal policy defined over the partition of the state space that minimizes (maximizes) some performance objective. Algorithms presented to solve this problem for the infinite horizon undiscounted average cost case have largely been based on enumerative procedures. A heuristic solution procedure based on Howard's (1960) policy iteration method is presented.

Applications of Markov decision processes with restricted observations exist in networks of queues, retrial queues, maintenance problems and queuing networks with server control. A new application area is proposed in the field of information sharing to measure the value of information sharing in a supply chain under optimal control. This is achieved by representing a model of full information sharing as a completely observable Markov Decision process (MDP), while no information sharing is represented as an MDP with restricted observations. Solution procedures are presented for the general Markov Decision process with restricted observations. Heuristic solutions are evaluated against the optimal solution obtained via total enumeration. Both random Markov Decision processes and information sharing problems are studied. The value of sharing information in a two-stage supply chain system is studied. The influence of capacity, demand, cost and retailer policy on the value of information sharing is considered. Insight on the structure of the optimal policy with and without information sharing is provided.

STATE CLUSTERING IN MARKOV DECISION PROCESSES WITH AN APPLICATION IN INFORMATION SHARING

by

LAUREN MARIE BERRINGS

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

INDUSTRIAL ENGINEERING

Raleigh

2004

APPROVED BY:

Dr. R.E. King Chair of Advisory Committee	Dr. T.J. Hodgson Co-Chair of Advisory Committee
Dr. H.L.W. Nuttle	Dr. D.L. Bitzer

BIOGRAPHY

Lauren Marie (Berrings) Davis was born in Albany, New York on October 5, 1968 to Mary (Shearill) and Emanuel Berrings. She attended Rochester Institute of Technology where she received her Bachelor of Science in Computational Mathematics in 1991. Upon graduation, Lauren was awarded a GEM (Graduate Degrees for Minorities in Engineering and Science) Fellowship and attended Rensselaer Polytechnic Institute in the fall of 1991. Her Master of Science in Industrial and Management Engineering was conferred in December 1992.

Lauren was offered a full-time software engineering position at IBM and relocated to North Carolina in 1993. Lauren has held many positions within the Networking Hardware and Personal System Group divisions at IBM focusing on system integration and tools to support manufacturing. She currently works in the Integrated Supply Chain division where she has received three IBM Excellence awards for her role in the implementation of SAP across three manufacturing geographies.

Lauren began to pursue her doctorate degree at North Carolina State University in 1998 while continuing to work full-time at IBM. She is very active in her community, participating in IBM sponsored mentoring programs and community based tutoring programs. She was honored for her community service in 2003 and received a 'Dare to Make a Difference' community award from the Ebonettes Service club of Durham.

ACKNOWLEDGMENTS

There are many people who helped to make this dream a reality. I thank Dr. Russell King for his guidance, support and patience throughout this process. I thank Dr. Thom Hodgson for the many challenges he unknowingly (or knowingly) gave me. Although they frustrated me at times, the challenges enabled me to grow in my knowledge and approach to research. I can finally agree that my code can have errors. I thank Wenbin Wei, who is also doing research in this area, for the many 'information sharing' sessions we have held on the path to finishing our research.

The support my family and friends provided was invaluable. I thank my husband Tyrone for his love support and counsel. He has been an ad-hoc committee member challenging me to think outside the box. I thank my mom for all her prayers, words of wisdom and encouragement. I thank my Baptist Grove church family for keeping me in their prayers throughout this journey.

Many thanks to my IBM manager and teammates for picking up the slack while I was in class or at the library working on my research. Last but not least, without God none of this would have been possible. I look forward to the next challenge he presents me.

Table of Contents

List of Tables

List of Figures

Chapter 1 Problem Description

1.1 General Description

This research examines state clustering in Markov Decision processes, specifically addressing the problem referred to as Markov Decision process with restricted observations. The general problem is a special case of a Partially Observable Markov Decision process where the state space is partitioned into mutually exclusive sets representing some observable portion of the process. The goal is to find an optimal policy defined over the observable portion of the state space that minimizes (maximizes) some performance objective. The policy obtained is referred to as an implementable policy. The initial motivation for work on state clustering derived from the need to reduce the dimensionality of the state space for large-scale multi-stage inventory problems, thus enabling solutions of these problems to be obtained in a reasonable amount of time on a computer. We are motivated to continue work in this area due to our interest in its applicability for measuring the value of information sharing in a supply chain. Howard(1971), Kemney and Snell(1960), and Dietz(1983) provide conditions under which a cluster state can be created. Observability constraints for the MDP were added to reflect systems in which the entire state space is not visible to the decision-maker. In this situation, new policies are required to determine optimal control based on what could be observed and implemented by a decision-maker in a real time system.

Algorithms developed to address this problem have covered infinite and finite time horizons as well as discounted and undiscounted costs. Serin and Avsar (1997) studied the finite horizon discounted cost case and proposed an algorithm that finds a global deterministic optimal policy. This research will cover the infinite horizon undiscounted average cost case, which has relied on heuristic procedures for determining implementable policies. This research will extend the work done by Smith (1971), Hordijk and Loeve(1994), and Hastings and Sadjadi (1979), to provide stronger bounds on local optimal solutions. Empirical results demonstrate that 90% of the problems generated can be solved to optimality, and the instances in which an optimal solution can not be found have an average error of 1%.

Applications of Markov decision processes with restricted observations exist in networks of queues, retrial queues, maintenance problems and queuing networks with server control (Hordijk and Loeve (1994),Serin and Avsar(1997)). A new application area is proposed in the field of information sharing. Specifically, the algorithm can be used to measure the value of information sharing in a supply chain under various supply chain structures. Information sharing entails sharing key pieces of operational data between supply chain partners to improve performance. Supply chain partners can share any combination of inventory, demand, sales forecast, and production or delivery schedule information. Supply chain members consist of suppliers, manufacturing sites, distribution centers, retailers and consumers which can be contained within one company or consist of several external parties whose resources are combined into an end product for the consumer. By sharing information, it is believed that the negative impact of uncertainty (demand, production, etc.) on the supply chain performance can be minimized.

Several applications of information sharing have been incorporated into the logistics operations of many companies, reportedly improving the efficiency of the members in the supply chain. Lee and Whang (2000) provide a thorough description of the levels of information sharing and the companies using such programs to improve supply chain performance. Several papers have been published quantifying the value of information sharing to the supply chain and characterizing the conditions in which it is most beneficial. These papers examine impact of sharing inventory, point of sale (POS) data, sales forecast, and production and delivery schedule data in several supply chain structures. The value of the information is measured by developing a model that compares the supply chain costs and order decisions with and without the additional information, and analyzing key performance measures such as average inventory, order quantity, backorders, and per period costs. The following assumptions are commonly made in these models.

1. Two-stage supply chains with a single supplier and retailer or a single supplier and multiple retailers
2. Capacitated supplier
3. Independent and identically distributed demand

4. Order-up-to or (s, S) inventory control policy employed by retailer and/or supplier
5. Periodic-review inventory management

For these models, optimal or near-optimal inventory control or supply allocation policies are determined and the cost savings associated with the various levels of information sharing compared. The methodology used to determine the optimal policies and cost benefits have varied. A gradient-based simulation procedure known as infinitesimal perturbation analysis is used by Zhao and Simchi-Levi(2002), Gavirneni (2001,2002), and Gavirneni *et al*.(1999). Zhao *et al*. (2002a) uses a simulation model to quantify the value of information sharing. Analytical models incorporating information flow into inventory control models are employed by Cachon and Fisher (2002), Lee *et al*. (2000), Yu *et al*. (2001) and Raghunathan (2001).

In this research, Markov Decision models are developed to determine the gain and optimal control policies, which are used to determine the associated savings with and without information sharing. A Markov model is a natural way to represent a system where information is shared. Based on the supply chain structure being used, the definition of the state space indicates the available information known to the decision-maker at any point in time. A completely observable MDP is used to model the information sharing case. The case of no or limited information sharing is modeled as a Markov Decision Process with restricted observations and solved via the algorithm proposed in this research.

1.2 Proposed Research

The research proposed is two-fold; to provide a new heuristic for solving the restricted observation Markov Decision problem and using it to analyze the value of information sharing under steady state optimal control. Randomly generated problems will be analyzed to validate the performance of the algorithm under different problem structures. MDP models for representing information sharing as a Markov Decision process will be developed for various supply chain structures and information sharing policies. A subset of those models will be analyzed to characterize the structure of the optimal policies under demand and inventory information sharing.

Chapter 2 contains a summary of the approaches taken to study the value of information sharing and the results gleaned from those models. Solution procedures and conditions for state clustering in Markov decision processes are also presented.

A new heuristic for solving the MDP with restricted observations (ROMDP) is outlined in chapter 3. Initial results are presented in chapter 4 for a simple two-stage supply chain sharing demand and inventory information. Results are also given for a randomly generated Markov Decision process. Sensitivity analysis and refinement of the heuristic are discussed in Chapter 5. Chapter 6 outlines a successive approximation counterpart to solving the ROMDP. The value of information sharing in a two stage supply chain is studied in Chapter 7. Experimentation considers the influence of variance, capacity, retailer inventory control policy and cost. Conclusions and further research are discussed in Chapter 8.

Chapter 2 Literature Review

2.1 Information sharing

2.1.1. Information Sharing Policies

Information sharing is the action of taking imperfect information and making it 'nearly' perfect. This action between supply chain members involves sharing one or more characteristics about the demand or manufacturing process. This additional information enables the recipient to provide better service in the form of better supply commitments, fewer lost sales or backorders, or better management of demand fluctuations and thus improved reliability. Some of the characteristics that can be shared are the actual demand realized during the period, the demand forecast, inventory position or inventory control policy. The parameters that define an information sharing policy are typically the type of information being shared and the direction of information flow between the participating supply chain members.

There is no standard terminology or nomenclature to characterize information sharing policies. The definition of the policy structure is usually subject to the researcher. However, the *no information sharing* policy is commonly recognized as the historical method of communication between suppliers and retailers. Under this policy, information about the retailer's demand or ordering policy is unknown. The only information the supplier receives is in the form of orders from the retailer. Therefore, the retailer's process is like a 'black box' to the supplier. Gavirneni(2001) uses the term partial information sharing to denote a policy in which the demand distribution and the parameters of the inventory control policy used by the retailer are known. Gavirneni(2001) also defines a full information sharing policy as one in which the demand distribution, retailer inventory control policy parameters and immediate demand information is known. A similar policy, with a different name is employed by Yu *et al.* (2001) to define a policy for vendor managed inventory.

The direction of information flow between supply chain members has traditionally been upstream in the form of orders. In a multi-stage supply chain structure, the lowest level is considered the point at which demand occurs. With new information

sharing policies being developed, the direction of information flow can continue to be upstream. The potential amount of information flowing upstream is now greater. For example, the supplier can have access to the retailer's inventory data in addition to point of sale data. The information can also flow downstream from the supplier to the retailer. An example of this type of policy may be in the form of consignment, where the retailer has access the supplier's inventory and is only charged based on the amount they extract from inventory. Bi-directional type of information flow can also occur between the supply chain members. Cachon and Fisher (2000) model this type of policy with 1 supplier and *N* retailers sharing inventory data between all members in the supply chain retailer to retailer as well as supplier to retailer. This type of supply chain configuration may be seen between regional distribution centers, where stock can be reallocated between the warehouses by a single controlling entity.

2.1.2 Models of information sharing

A vast amount of literature has been published studying the value of sharing various types of information within different supply chain structures. The information shared is largely inventory, demand or point of sale data. Huang *et al*. (2003) provide a comprehensive summary of the literature covering this topic, as well as a theoretical framework for future research. Several supply chain structures are defined and used as a hierarchy for categorizing existing information sharing research. A dyadic structure, depicted in Figure 2.1, represents a two-stage supply chain consisting of a one-to-one relationship between two business entities. A partnership between a supplier and retailer is an example of a typical dyadic structure. A serial supply chain represents an *N* stage structure where each supply chain member performs its activity sequentially. A typical serial supply chain may consist of a supplier, manufacturer, distributor and retailer. An example of a serial supply chain is depicted in Figure 2.2. A divergent supply chain, shown in figure 2.3, represents a two-stage network consisting of a single entity supplying several parallel entities. A distribution chain consisting of a single supplier and multiple retailers is a typical example of a divergent structure. A convergent structure, as shown in figure 2.4, is a variant of a serial structure. A manufacturing supply chain typically represents a convergent structure, where you may have multiple

suppliers and multiple stages in the manufacturing process depending upon the structure of the end item being produced. Finally, a network structure is a complex supply chain combining elements from the divergent and convergent structure. Refer to figure 2.5 for an example of a network supply chain structure. The bulk of the research consists of analytical models of dyadic and divergent structures. Selected papers relevant to this research are discussed in the following sections and summarized below in table 2.1 by supply chain structure and modeling approach. Some of the models are included in the hierarchical summary by Huang *et al.* (2003) and some are new contributions since the publication of his work.

Table 2.1 Information Sharing Research By Structure and Model

SUPPLY CHAIN STRUCTURE	SIMULATION	GAME THEORY	ANALYTICAL	MATHEMATICAL PROGRAMMING
Dyadic	Gavirneni *et al.* (1999)		Lee *et al.* (2000) Raghunathan (2001) Yu *et al.* (2001) Gavirneni (2002) Zhao and Simchi-Levi (2002)	
Divergent	*Zhao* et al. *(2002a)* *Zhao* et al. *(2002b)*	Li (2002)	Cachon and Fisher (2000) Gavirneni (2001)	
Serial			Chen (1998)	
Convergent			Wei and Krajewski (2000)	
Networked				D'Amours *et al.* (1999)

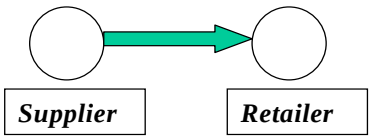


Figure 2. 1 Dyadic Supply Chain

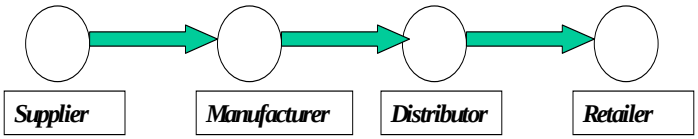


Figure 2. 2 Serial Supply Chain

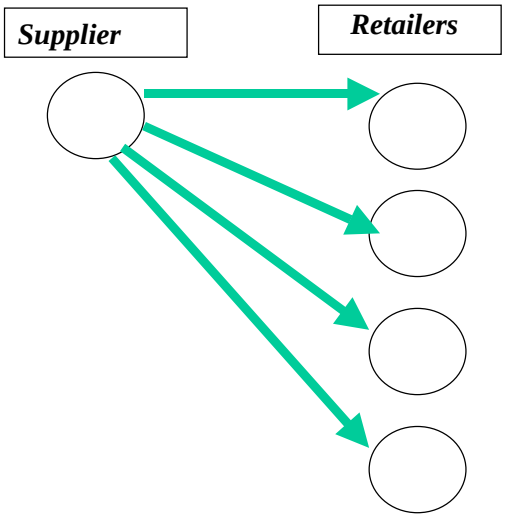


Figure 2. 3 Divergent Supply Chain

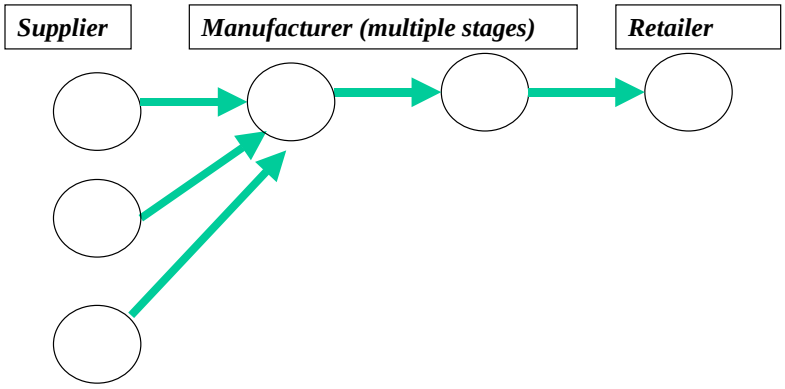


Figure 2. 4 Convergent Supply Chain

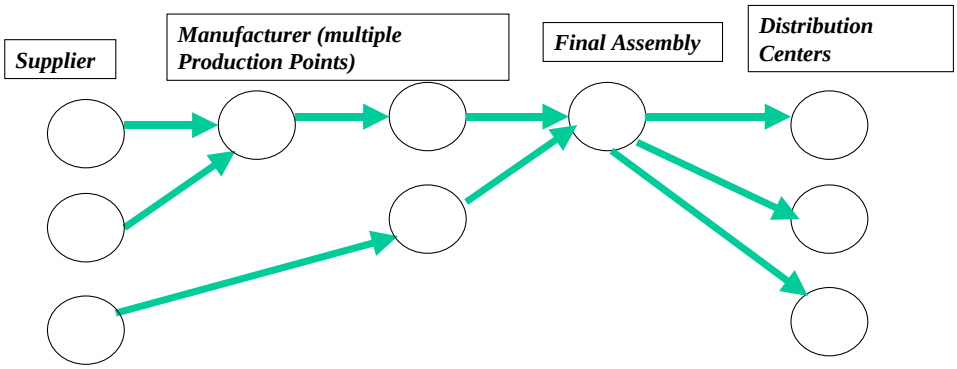


Figure 2. 5 Network Supply Chain

2.1.2.1 Simulation Models

Zhao *et al*. (2002a) use a simulation model to study the value of sharing sales forecast information in a divergent supply chain system consisting of a single capacitated supplier and four retailers. Transportation lead-time is one period, implying that a single truck can deliver the required shipment to each retailer during that time. The retailers use an EOQ inventory policy and the supplier uses single-item capacitated lot-sizing to plan its production activities. Backorders are allowed at the supplier and retailers. The cost savings to the supply chain are evaluated by varying retailer's demand pattern, supplier capacity, information sharing level, and order coordination. Ordering coordination (OC) refers to negotiating longer lead-time for parts by placing orders with the supplier in advance. Conditions by which the retailer and/or supplier benefit are characterized by examining the decisions made by the supplier under each information sharing policy and quantifying the resulting affect on the performance of the supply chain. When no information is shared (NIS), the supplier's production decisions are based on the orders received from the retailer. When demand forecast data is shared (DIS), the supplier's production decisions are based on the retailer's order and the forecasted demand. The supplier's decision under the policy of sharing planned orders (OIS) is based on the retailer's order and future planned orders generated as a result of the retailer demand forecast. Using the same model assumptions described above, Zhao *et al.* (2002b) also study the impact of forecast model selection on the value of information sharing. Several forecasting methods are evaluated ranging from simple models with poor level of accuracy to exact models representing a perfect forecast. Forecast accuracy is measured in terms of standard deviation of the forecast error. Simulation is used to evaluate the difference in total costs and service level when the Retailer uses different forecasting methods. They also examine the influence of the forecast model selection on the cost savings associated with the information policies (OIS,DIS, and NIS). The results indicate the value of information sharing is higher when the forecast accuracy is high.

Gavirneni *et al*. (1999) study the value of demand information sharing in capacitated supply chains. The supply chain model has a dyadic structure with periodic review, zero supplier lead time, independent and identically distributed retailer demand and an (*s, S*) inventory control policy for the retailer. The retailer always obtains the

order quantity either in full from the supplier or in part from the supplier and part from somewhere else. Infinitesimal perturbation analysis is used to determine the optimal order up to level and total costs incurred by the supplier at each level of information sharing. Simulation is performed to measure the value of information sharing under the optimal policies. The interaction between information sharing and different measures of performance, such as inventory and capacity, are also examined by varying demand distributions, levels of capacity, and inventory control parameters. The three possible policies associated with information sharing are no sharing, partial sharing and full sharing. Under a policy of no information sharing, information about the retailer demand or ordering policy is not known. The supplier demand and subsequent control policy is based on the order quantity. With a partial information sharing policy, the demand distribution and the parameters of the inventory control policy are known. With this information, the supplier can determine the probability of an order being generated at the end of the period and the CDF of the order size. Under a policy of full information sharing, the demand distribution, (s, S) policy parameters, and immediate information about demand are known. Again, the CDF of the order size and probability an order is placed can be determined. The results indicate increasing levels of information flow, in all cases, reduces the supplier's costs. The degree of the savings depends on the capacity available, the end-item demand variance, and the retailer order quantity (S-s).

2.1.2.2 Analytical Models

Lee *et al.* (2000) develop a base stock model to investigate the impact of sharing point of sale data in a two-stage supply chain consisting of a single retailer and a single manufacturer. The retailer demand follows a first order autoregressive (AR(1)) process and both the retailer and manufacturer employ an order-up to inventory control policy. The ordering cost is assumed to be zero and the manufacturer knows the demand follows an AR(1) process. When no information sharing (NIS) occurs, the manufacturer's order decision is based solely on the demand as a function of the retailer's order quantity at the end of the period. Under information sharing (IS), the manufacturer receives the order quantity and the retailer's demand at the end of the time period. Based on the information being shared, the supplier's order up to level as a function of the forecast demand can be determined. Expressions to quantify the average inventory and expected

cost as a function of the forecast demand are developed. The variance of the forecast demand is smaller when information sharing occurs and thus the supplier experiences inventory reduction. The expressions obtained analytically are verified with a simulation study.

Raghunathan (2001) uses the model developed by Lee *et al*. (2000) to demonstrate the value reported is insignificant. The key difference between the two models is that under a policy of no information sharing, Raghunathan assumes the retailer order history is used to forecast future orders while Lee *et al*. (2000) assume the manufacturer uses only the most recent order from the retailer to forecast future orders. Ragunathan reports the value of information sharing decreases monotonically with each time period, converging to zero in the limit. He suggests that information sharing of demand data can be valuable to the manufacturer if none of the demand parameters can be inferred from the order history.

Yu *et al*. (2001) analyze the value of sharing point of sale and inventory information in a two-stage supply chain. They develop a discounted cost-minimizing inventory model that is used to derive the optimal inventory policy for the members in the supply chain. The resulting policy is then used to analyze the average inventory level and expected costs under the different levels of information sharing. Both supply chain partners use an (*s*, *S*) inventory control policy with periodic review. Excess demand is backlogged and each supply chain member incurs holding, penalty, and order costs during each period. The information sharing and order coordination policies evaluated are no sharing, coordinated control and centralized control. When no information is shared, the inventories at the different sites are controlled independently. Under a policy of coordinated control, the retailer's customer demand is shared with the manufacturer. The manufacturer’s order decision is based on both the customer demand the retailer's order information. When complete information and coordination occurs (centralized control), the customer demand and retailer inventory information is shared. The manufacturer uses Vendor Managed Inventory (VMI) policy to coordinate replenishment at the retailer.

Chen (1998) examines the value of using localized demand information versus centralized demand information in a multi-stage production system. The system incurs

linear holding costs at every stage and linear backorder costs at the first stage. Production lead times are constant between stages and it is assumed that a reorder point /order quantity policy is used at all stages. The order quantities at each stage are fixed and the reorder point at each stage is the decision variable. The value of centralized demand information is measured as the relative cost difference associated with implementing an echelon based batch reorder point policy and installation based batch reorder point policy. Echelon based policies represent the optimal replenishment strategy when centralized demand information is used, while installation based policies represent the optimal replenishment strategy when local demand information is used. The optimal echelon reorder point policies are determined by decomposing the problem into single-stage models which are solved sequentially, similar to the approach developed by Clark and Scarf (1960). The installation based reorder point policies are determined using a bounded search procedure.

Cachon and Fisher (2000) develop an analytical model to examine the value of inventory information sharing on a supplier's order and allocation decisions in a periodic review system. The supply chain structure consists of *N* identical retailers and a single infinite capacity supplier. Batch reorder point policies are used by the retailer and the supplier when no information is shared and by the retailer only when information is shared. The supplier's optimal policy and allocation decision are determined from the shared information. Under traditional information sharing, the supplier receives only the order quantity and allocates available inventory based on a batch priority scheme. In full information sharing, the supplier knows the inventory level at all retailers, and allocates supply in a manner that balances the retailer's inventory levels across the system. The additional shared information is used to determine the optimal policy and allocation decision for the supplier. Simulation is used to approximate the optimal policy and estimate the expected per period supply chain costs associated with the full information sharing policy. The optimal policy and per period costs under the traditional case is determined via a search over all feasible policies.

Gavirneni (2001) also examines the value of inventory information sharing on a supplier's allocation decision in a divergent supply chain environment. The model is similar to that of Cachon and Fisher (2000) with the following exceptions: the supplier is

capacitated; there is no batch size for reordering; and the benefits of information sharing are computed by comparing optimal policies. The model of Cachon and Fisher (2000) uses at least one non-optimal policy in computing the benefits. When no information is shared the orders are filled in a predetermined sequence. When partial information is shared, retailer demand and inventory levels between the supplier and retailers are known. Inventory is allocated amongst the retailers in a manner that ensures retailers with lower inventories receive larger shipments. In a system with complete information sharing, inventory levels are shared between all members in the supply chain, supplier to retailer and retailer-to-retailer. Retailers with very high inventory levels are willing to give up inventory and face higher penalty costs in order to help those retailers with very low inventory levels. As a result, the supplier can move inventory between retailers to satisfy other retailers order quantities. Demand not satisfied by the supplier is lost and demand not satisfied by the retailers is backlogged. Infinitesimal Perturbation Analysis is used to compute the optimal order up to level and per period retailer holding and penalty costs for each model. When no information is shared, each retailer has its own order up to level, while in the other models each retailer has the same order up to level.

Gavirneni (2002) examines the effect of information sharing when operating policies used by the retailer are changed to make better use of the information flows within the supply chain. Two models consisting of demand and cost information sharing between a supplier and retailer are evaluated. In the first model (Model 1), the retailer uses his optimal (s,S) inventory control policy and the optimal order up to policy for the supplier is determined using IPA. The supplier knows the cumulative demand at the retailer since the last order occurred and therefore can estimate, in each period, the probability that an order will occur and the CDF of the order size. In model 2, the retailer places orders after his next customer demand only if his current cumulative demand in the period is greater than some value, δ. The optimal δ value must be found using an exhaustive search. The resulting policy used by the retailer in model 2 is an order-up-to policy. This model allows the supplier to know a period in advance that an order is being placed and again he can estimate the CDF of the order size. Under the assumptions of model 2, the supplier knows for certain that an order is occurring in the next period. That information is not known with certainty in model 1and therefore holding costs may be

incurred as a result of producing earlier in anticipation of demand. The cost function is formulated as a stochastic dynamic program and IPA is used to compute the optimal order up to levels.

Zhao and Simchi-Levi (2002) also use infinitesimal perturbation analysis (IPA) to quantify the cost savings to the supplier when demand information is shared in a two-stage production inventory system. The information sharing problem is modeled as a Markov decision process to prove that a cyclic order up to policy for the supplier is optimal and has a finite steady state average cost for the discounted and average cost criteria. However, infinitesimal perturbation analysis is used to compute the optimal policy and compare the resulting costs under the different information sharing levels. The retailer uses a periodic review system with an order-up-to inventory policy. The model also examines the effect of frequency and timing of information shared on the costs incurred by the supplier. The point in time where demand is shared and production decisions can be made but no retailer order is placed is referred to as an Information Period. The time at which retailer orders are placed is referred to as an Ordering Period. Several information periods exist between ordering periods. Under a policy of no information sharing, demand information is received at the order interval. When information sharing is employed, the retailer shares point of sale data for each information period and places orders during their ordering period. The model of no information sharing is the information sharing model with zero information periods.

Wei and Krajewski (2000) examine the value of sharing schedule information in a convergent supply chain structure. There is a single manufacturer with several tiered suppliers. The first and second tiers of the supply chain provide lower level components for the production of an end item. The manufacturer must try to determine the best scheduling policy based on the information available. The information available is determined from the level of integration between the manufacturer and suppliers within the supply chain. There are several suppliers within the supply chain structure at different tiers with whom the schedule information can be shared. The level of sharing and thus integration of the schedule within the supply chain evaluated are Myopic, Tier – 1, Critical path and total. Under a myopic level of integration, the manufacturer only uses its cost when determining its scheduling and purchasing policy. With tier-1

integration, the manufacturer considers the flexibility capability of all tier 1 suppliers when determining his scheduling policy. Under critical path, only the capabilities of the critical path suppliers are used in the determination of the best policy. Total integration considers the capability for all members in the supply chain. Flexibility capabililty is a numerical measure describing the ability of the members to adapt to schedule changes by the manufacturer. A stochastic cost model is used to determine the best policy that minimizes the total costs associated with schedule changes, shipping costs, material and inventories. Results indicate that the ranking of costs associated with schedule integration is total sharing < critical path < tier-1 < myopic. Schedule change costs are a primary driver of the optimal policies and the cost ranking is unaffected by demand variation. Based on analysis of the results, Wei and Krajewski (2000) suggest that it is more cost effective for the manufacturer to focus on integration with suppliers in the critical path.

2.1.2.3 Game Theoretic Models

Li (2002) models the single manufacturer - multiple retailers supply chain as a Cournot competition game. The objective is to study the direct effect and indirect effect of demand information sharing in a supply chain. The direct effect consists of the payoff achieved between the partners directly involved in the information sharing. The indirect effect consists of the effect on other competing firms due to leakage of information. As a result of the information leakage, the competing firms may respond by changing their ordering strategies. The leakage effect occurs by assuming that the supplier's product price is a monotone function of the sum of the shared retailer's information. Therefore, retailer's can infer the sum of the shared signals from the supplier's price. The model consists of a 3-stage subgame. During the first stage, the retailer's must decide whether to share their private information with the supplier. At the second stage, the manufacture sets the price for the product being supplied based on the information known about the demand. At the last stage, the retailers choose their sales quantities. From this model, expressions are derived for the expected equilibrium profits and equilibrium sales quantities from which the value of information sharing can be quantified.

2.1.2.4 Mathematical Programming Models

D'Amours *et al*. (1999) use a network flow model to examine the impact of price and capacity information sharing in a networked manufacturing environment. The supply chain structure consists of a single (networking) firm with partnerships between several manufacturing, transportation and storage firms. The networking firm must choose and schedule the order among the available firms in order to satisfy the customer order. The information policies are expressed in terms of the bidding protocols representing information transferred between the networking firm and the contracting firms. In supplier–type bids, information transferred in the bid is publicly known price and time packages. In customizing-type bids, price and time packages are customized based on the needs of the networking firm. The networking firm shares capacity and time requirements. The contracting firms share price-time package information based on needs of the networking firms. The package represents a maximum set of alternatives that can be constructed to support the order. In webbing-type bids, information shared from contracting firms to networking firms is day-to-day operating characteristics, production capability, capacity requirements and pricing functions. From this information, the networking firm generates their own set of bid alternatives. All possible bids within each type are formulated as a network flow problem. The objective is to configure and schedule a virtual manufacturing and logistics network to satisfy delivery and quantity requirements of customer.

2.1.3 Value of sharing information

The existing models examine the value of information sharing from different reference points; the supplier, the retailer, and the total supply chain. Actual results on the value associated with sharing information differ based on the assumptions of the model, reference point, and integration of the information into the decision process. As a result, it is difficult to understand how and when information should be shared, and how to develop a best practice within the supply chain as a result of undertaking information sharing partnerships. The model assumptions, capacity, demand distribution and measure of performance influence the results from the models discussed in section 2.1.1. One common thread amongst all results is that the supplier clearly benefits in all cases of information sharing. Additional information creates better demand information resulting

in reduced inventories and reduced costs. A summary of the results is presented by supply chain structure and reference point.

2.1.3.1 Benefits to the supplier under Dyadic Models

Lee *et al*. (2000) show the inventory reduction and cost reduction with information sharing is significant only when the demand is highly correlated over time, highly variable or when the lead-time is long. However, Ragunathan (2001) shows, for the same model, that information sharing decreases monotonically with each time period when the supplier uses a better forecasting method under no information sharing. Only when the demand parameters cannot be inferred from the order history, is information sharing significant. These two papers demonstrate how results vary based on the integration of information into the decision process. Gavirneni *et al*. (1999) also examine the effect demand variance has on possible savings with information sharing. They varied the variance of Normal, Uniform, and Erlang distributions while keeping the mean value common. For each distribution, the percentage savings increased and then decreased as the coefficient of variation was decreased. They concluded the variance of the retailer's demand distribution limits the cost benefits that can be achieved with information sharing. When the demand variance is high, the reduction in uncertainty due to the additional information is insignificant from a cost perspective. At moderate values of demand variance, information sharing appears to be most beneficial.

Gavirneni *et al*. (1999) also examine the effect capacity has on the value of information sharing. Under low levels of capacity cost benefits are not significant. However, as supplier capacity increases, some reduction in cost can be achieved due to the fact that information sharing allows the supplier to postpone production. Zhao and Simchi-Levi (2002) also demonstrate significant supplier cost savings as capacity increases. Their computational study illustrates savings ranging from 5 to 35 percent as production capacity increases. In addition, percentage savings increase as the number of information periods within an ordering period increase. However, most of the benefit is achieved within a few information periods. In terms of the timing of information sharing, they conclude that when capacity is large relative to mean demand, it is appropriate to postpone the time of information sharing to the last production opportunity in the

ordering period. When capacity is tight, the cost is less sensitive to the timing of information sharing.

With respect to the inventory control parameters of an *(s,S)* policy, Gavirneni *et al*. (1999) show the percentage savings between partial information sharing and full information sharing has no significant differences. Both policies demonstrate the information is less beneficial at extreme values of the order quantity (*S*-*s*). The authors attribute this behavior to the fact that the extreme order quantities reduce the benefit of sharing information. When (*S*-*s*) is large, the supplier has to build up inventory in anticipation for a large order. When (*S*-*s*) is small, the demand information is passed to the supplier almost every period, thus reducing the benefit of sharing demand information.

Yu *et al*. (2001) use their model to study the expected per period costs. The results show that suppliers benefit under each increasing level of information sharing; no information sharing, coordinated control (demand is shared) and centralized control (demand and inventory is shared). As information is shared, the inventory levels at the manufacturer decrease, resulting in smaller expected per period cost.

2.1.3.2 Benefits to the Retailer under Dyadic Models

Yu *et al*. (2001) examine the affect of information sharing on the retailer as well as the supplier. The key results indicate there is no benefit to the retailer by sharing their customer demand with the manufacturer. The average inventory and expected costs between coordinated control (demand information is shared) and no information sharing remain the same. Under centralized control, where demand and inventory information is shared, the retailer realizes performance improvement because the retailer's lead-time is reduced due to the improvement in the manufacturer's reliability as a result of using VMI. Since the supplier receives most of the benefit from information sharing, the authors suggest some incentive should be offered to induce the retailer to share their demand information.

2.1.3.3 Benefits to supply chain partners under Divergent models

When competition and information leakage is incorporated in the model, Li (2002) shows the supplier is better off acquiring information from as many retailers as

possible. However, the expected profit for the retailers is less when they share information. The incremental loss from sharing gets smaller as more retailers share information. So the resulting equilibrium strategy is to not share information. From a total supply chain perspective, profit is larger with information sharing when the information each retailer has is informative in a statistical sense or when there is a sufficiently large number of retailers. Li also suggests the manufacturer should provide incentives to the retailer to share information and discusses a contract signing game where retailers are compensated by some fixed amount. Boundary conditions are provided for the compensation value.

Zhao *et al*. (2002a,) also study the value of information sharing from the supplier, retailer and total supply chain perspective. The supplier is usually the benefactor in all cases of increased information sharing and ordering coordination. The retailer benefits only when all retailers face identical demands with decreasing trend and the supplier's capacity utilization with respect to capacity needed to meet the demand is high (85% or 95%). When order coordination is high and demand is different for each retailer, the supplier's service level increases but at the expense of the retailer and the supply chain. High order coordination implies the lead-time between orders increase. Retailers are placing their orders several periods in advance. When this occurs, retailer's forecast errors increase, resulting in deteriorated service levels and increased backorder costs. Therefore, total costs for the retailers and the supply chain increase. Similar results are reported when studying the impact of forecast model selection on the value of information sharing. (Zhao *et al*. (2002b)). In addition, the value of information sharing is greatest when the forecast accuracy, measured in terms of standard deviation of the forecast error, is high.

Under varying levels of capacity, Zhao *et al*. (2002a) illustrate sharing planned orders performs better than sharing no information or demand information. Information sharing is beneficial to the supplier under all levels of capacity tightness (the ratio of the total available capacity to total capacity needed). The total capacity needed is a one-to-one relationship with the demand requested. The authors suggest the supplier's benefit is due to his ability to make better use of his capacity and fill more retailers' orders in time. However, Gavirneni (2001) analyzes the value of information sharing from supplier cost

view and finds that information is more beneficial at lower capacities and higher penalty costs, when comparing no information sharing to demand and inventory level sharing. At lower capacities, the supplier is not able to meet all of the retailers' demands and information enables him to allocate capacity better. When capacity is high, the demand can be satisfied for all retailers and thus information is not beneficial. Gavirneni (2001) measures capacity as a ratio of supplier's capacity to the retailer's mean demand. This contradicts results discussed earlier in Gavirneni *et al*. (1999) for a dyadic supply chain structure. However, these results demonstrate how the affect of capacity on the value of information sharing differs under different supply chain structures, information sharing policy, measures of capacity and model assumptions.

Gavirneni (2001) also finds information to be less beneficial between no information and some cooperation (demand and inventory sharing) when demand is not highly variable. Using the Erlang and Exponential distributions, results show as variance decreases, the percentage benefit decreases. He also studies the affect the number of retailers in the distribution chain has on the value of information. In this case, as the number of retailers increase, the benefit from information sharing decrease.

Cachon and Fisher (2000) look at total supply chain costs to quantify the benefits of information sharing. Their results indicate full information sharing provide an average benefit of 2.2% over the traditional information sharing case. The authors also conclude that higher cost savings can be achieved through lead-time reduction and smaller order batch sizes. Lead time reduction results in an average cost savings of 21% while batch size reduction results in an average cost savings of 22%.

2.1.3.4 Benefits to supply chain under Network models

D'Amours *et al*. (1999) examine the impact of sharing capacity and price information in a networked based supply chain. Using network flow analysis, they determine that web-type bids (day-to-day production capability and capacity information is shared) achieve the lowest cost network with a cost reduction of 28.2% over the standard supplier-type bids (price time information is shared). The results show that as the contractors share more information on the price and capacity with the networking firm, better price-time scheduling performance is achieved. However, better price-time

scheduling comes at the cost of more complexity. The complexity is in terms of the number of manufacturing and logistics units selected to schedule the order.

2.1.4 Markov modeling approach to information sharing

In all of the papers studying information sharing, no one has examined this problem from the perspective of steady state optimal control. A Markov model is a natural way to represent a system where information is shared. Based on the supply chain structure being used, the definition of the state space indicates the available information known to the decision maker at any point in time. By collapsing the state space, you restrict the information known and affect the policy chosen. Both models yield the steady-state optimal policy and gain, which provide a consistent and equivalent measure of performance between the two systems. There are several advantages to studying information sharing as a Markov Decision process. There is a single model that yields the optimal policy for the decision maker along with one consistent measure of performance: the gain. With simulation, you are not comparing the systems under optimal environments. You are approximating the performance of near-optimal policies. In the area of Markov Decision problems, there has been extensive research discussing methods for rapid convergence and computational efficiency (e.g. Ding *et al*., 1988 and White, 1963) to assist in studying large-scale problems. Therefore, the existing research in supply chain information sharing can be extended to study larger and more complex supply chain structures. In addition, it is easy to analyze information sharing from different vantage points by structuring the costs from the desired view; total supply chain, retailer, or supplier. The next section introduces the recent work in state clustering in Markov decision problems, which enables us to take the completely observable Markov process and restrict it to a partially observable process, representing a system with no or limited information sharing.

2.2 Markov Decision Processes

2.2.1 State clustering

Early work on state clustering in Markov Processes was motivated by reducing the dimensionality of large problems to make them reasonably solvable with a computer.

Howard (1971) defines conditions under which the states of a Markov process can be grouped to define a new state, referred to as a 'super-state'. The resulting process with super-states is called a Mergeable Markov Process. The partitioning of states into super-states is dictated by the transition probability. Each member in a super-state must have the same probability of transitioning to another super-state. Kemney and Snell (1960) formally define this condition for the finite horizon Markov Chains as strong lumpability. For every pair of super-states, S_k and S_j,

$$p_{iS_j} = \sum_{k \in S_j} p_{ik} \quad \forall i \in S_k \tag{2.1}$$

where p_{ik} is the transition probability from state i to state k, and $p_{iS_j} = p_{kS_j} \; \forall i, k \in S_k$. The new states are now represented by the sets formed from the merged process with transition probabilities defined by equation (2.1). The benefit of a mergeable process is that it allows very large problems to be scaled to a more manageable size and solved using existing Markov Chain theory, thus, allowing for analysis on the state groups as opposed to the original states.

Dietz (1983) described similar conditions for a strongly lumpable Markov Decision Model. Given two countable sets E and $E`$ where, $|E| \geq |E`|$, ϕ is defined as a function mapping E -> $E`$. This function represents a cluster mapping or lumpation of the state space $E$, where $\phi(x)$ is an element of $E`$ and is a cluster state, and x is an element of E. Given a cluster state s in $E`$, $\phi^{-1}(s) = \{ x \in E$ , where $g(x) = s : g(\,) :=$ cluster mapping $E$ to $E`\}$. The ExE transition matrix P is strongly lumpable if it satisfies equation (2.1), which implies $P(x, \phi^{-1}(s)) = P(y, \phi^{-1}(s))$ holds for all x,y in E with $\phi(x) = \phi(y)$. If P allows a strong lumpation, then an $E`$x$E`$ transition matrix $P`$ is defined in the same manner described by Howard. A Markov Decision model associated with the lumped transition matrix P` has a decision space D` defined for the cluster image such that the decision function for cluster images: $D`(s) = D(x)$ if $\phi(x)=s$ for s in E` and $\phi(x)= \phi(z)$ implies $D(x) = D(z)$ for all $x$, $z$, in $E$. This states the decision space must be identical for all states in the cluster state. A payoff structure $r`(s,d)$ for the cluster image defined for s in $E`$, $d$ in $D`(s)$, is equivalent to $r(x,d)$ if $\phi(x)=s$ holds. Although not stated, this implies the reward value is the same for all states x in the cluster state s. The recursive

relations developed by Howard (1960) hold for the lumped chain and can be used to find an optimal policy for the lumped process for a finite horizon problem.

In the algorithm proposed in this research, we are not requiring the transition matrix to exhibit strong lumpability, nor are we redefining the transition matrix in terms of the state groups. We are analyzing the original process, but restricting the set of feasible policies to those that are applicable to a super-state. The policy constraint requires all states in a given superstate to have the same optimal action. Thus, the optimal policy is defined based on the superstate (or set partition) and not the individual states of the Markov process. In this context, the superstate can be interpreted as an observable part of the Markov decision model under analysis.

2.2.2 Markov processes with partial information

Smallwood and Sondik (1973) study Markov decision processes with partial information, both in the finite horizon and infinite horizon case. In a Partially Observable Markov Decision Process (POMDP), the internal state of the system cannot be directly observed. However, some output of the system, θ, is observable and is probabilistically linked to the true state of the system. These observed outputs are used to determine the true state of the system. Along with the observed outputs, there exists a set of alternatives from which the optimal control alternative is to be determined. If the prior state of information about the internal sate of the system is denoted as π and we observe output θ after using alternative *a*, then the updated probability that the internal state of the system is *j* given the new information is $\pi'_j = \sum_i \pi_i p_{ij}^a r_{j\theta}^a / \sum_{i,j} \pi_i p_{ij}^a r_{j\theta}^a$.

Each output and control alternative determines a different vector in the space of information vectors and therefore π acts as a continuous state, discrete time Markov decision process where the state space is all possible π vectors. The optimal control alternative at a point in time is a function of all possible distributions of the π vector. Smallwood and Sondik (1973) devise a dynamic programming algorithm for the finite horizon case over the space of information vectors, which is similar to Howard's (1960) dynamic programming formulation for the completely observable process. The payoff function is defined similarly as:

$$V_n(\pi) = \underset{a \in A(n)}{Max}\left[\sum_i \pi_i\left(q_i^a + \sum_{i,j,\theta} p_{ij}^a r_{j\theta}^a V_{n-1}\left[T(\pi \mid a, \theta)\right] \quad \right)\right]$$

This expression defines the expected reward the system can accrue if the current information vector is π and *n* control intervals are remaining. It is determined from the immediate reward (q_i^a) associated with being in state *i* plus the expected reward if the system transitions to state *j* and observes output θ with one fewer control interval remaining. This function is piecewise linear, convex and partitions the space of information vectors into regions where one alternative is the maximizing alternative for all vectors in that area.

For the infinite horizon case discounted cost case, Sondik (1978) develops an algorithm to find near-optimal control alternatives. The method finds a set of Markov Partitions, the associated control functions for each partition, and the markov mapping defined for each observation and partition. A Markov partition is set of information vectors that have the same control alternative and the same markov mapping. A Markov mapping is a function that defines which partition or set of information vectors the system is likely to transition to when output θ is observed. Therefore, if the decision maker knows what set he starts in and can observe some output, the Markov mapping indicates the next set of information vectors the system will transition to, which in turn indicates the next control alternative to operate under. The control alternatives found may not necessarily be deterministic.

The optimal control policies for partially observable processes are often randomized policies mapped against all possible states of the information vector. These policies are difficult to implement in practice. The algorithm proposed in chapter 3 will determine the optimal deterministic policy associated with the observable outputs. The class of problems which maps the policy set to the observable outputs is known as the Markov Decision process with Restricted Observations.

2.2.3 Markov processes with restricted observations

2.2.3.1 General Problem

A Markov Decision Process with restricted observations is a special case of a POMDP. For a POMDP, the state of the system cannot be directly observed. However,

some output of the system is observable and is probabilistically linked to the true state of the system. These observed outputs are used to convert the unobservable MDP with finite state space to an observable MDP with a continuous state space. The observability assumption for the Restricted Observation problem partitions the state space *S* into *K* sets $S_1, S_2, \ldots S_k$ which are mutually exclusive. Adopting the notation from Smallwood and Sondik, a matrix *R* of observable outputs consists of row vectors that sum to unity and defines the probability of observing output θ given the true state of the system is *j*. For the MDP with restricted observations, each row vector of the matrix *R* contains exactly one entry with value one if the state is in output set *k* and zero otherwise. Thus, mutually exclusive sets are created. The best policy found is implementable for the observed set and is not a function of the set of all possible distributions of the information state vector, π. The policy for the partitioned state space is called an implementable policy with respect to the partition *S*. In an implementable policy, every state that is a member of partition S_k takes the same action at time *n* (A_n) with the same probability. Formally, $P(A_n = a \mid X_n = i)$ is the same for all states $i \in S_k$.

Table 2.2 Research summary by method

Time Horizon	Discounted Total Cost	Undiscounted Average Cost
Finite	*Nonlinear Programming*	
Infinite	*Nonlinear Programming*	*Successive Approximation* *Enumerative Search* *Bounded Enumeration*

Table 2.3 Research summary by author

Time Horizon	Discounted Total Cost	Undiscounted Average Cost
Finite	*Serin and Avsar (*1997*)*	
Infinite	*Serin and Kulkarni* (1995)	*Smith* (1971), *Hordijk and Loeve* (1994), *Hastings and Sadjadi* (1979)

There have been several approaches to solving this problem both for the infinite and finite horizon instances. Table 2.2 summarizes this research by time horizon and cost function. Table 2.3 summarizes the corresponding research effort by author. The

notation listed below will be used in the following sections that discuss solution procedures implemented for this problem.

A: The set of available actions $\{1 \ldots M\}$.

S: The set of possible states $\{1 \ldots N\}$.

O: The set of observable outputs $\{1...K\}$.

G_i: A function mapping state i to a single observable output in the set O.

S_k: A given partition of the state space S satisfying $\{i: G_i = k\}$.

X_n: A random variable denoting the state of the system at time n=0,1….

Y_n: A random variable denoting the observation at time n taking on values in the set O.

A_n: Action chosen at time n.

α: The policy vector for the observed process $[\alpha_1,\alpha_2 \ldots \alpha_K]$ where α_k is the action chosen for each state in the observation set S_k and $\alpha_k \in A$.

Π: The vector of steady state probabilities, $[\pi_1,\pi_2,..\pi_N]$ of the Markov process, where π_i is the long term probability of being in state i.

$p_{ij}(a)$ The one step transition probability from state i to j under alternative $a \in A$.
$p_{ij}(a) = P\{X_{n+1} = j \mid X_n = i, A_n = a\}$

c_{ia} : The immediate expected reward associated with transitioning from state i under alternative $a \in A$. $c_{ia} = E\{C(X_n, A_n) \mid X_n = i, A_n = a\}$. In Howard's (1960) algorithm, this is denoted q_i^a.

g^α: The steady state gain associated with a policy α.

g^*: The optimal gain associated with an instance of the problem.

$p_j = P(X_0 = j)$ is the initial probability at time $n=0$.

2.2.3.2 Infinite Horizon Discounted Cost

Serin and Kulkarni (1995) propose an algorithm to find locally optimum policies for the infinite horizon discounted cost case. The algorithm is based on a nonlinear programming formulation of the discounted cost problem. The algorithm is not guaranteed to find a deterministic policy but gives sufficient conditions under which a deterministic global optimal policy exists. Serin and Kulkarni (1995) refer to the policies generated from the algorithm as implementable policies, which may be randomized or deterministic. The restriction of the policy set for a completely observable MDP to an implementable policy is achieved by introducing observability constraints in the linear

programming model developed in Derman (1970), Kallenberg (1983) and Ross (1983). The decision variable for the linear programming model, x_{ia}, represents the long run proportion of time the process is in state *i* under alternative *a* discounted by factor γ.

$$x_{ia} = \sum_{n=0}^{\infty} \gamma^n P\{X_n = i, A_n = a\} \qquad (2.2)$$

Determining the optimal policy that minimizes the expected total discounted cost over the infinite horizon is found by solving the problem below.

Minimize $\quad \sum_{i=1}^{N} \sum_{a=1}^{M} c_{ia} x_{ia}$

Subject to $\quad \sum_{a=1}^{M} x_{ja} - \gamma \sum_{i=1}^{N} \sum_{a=1}^{M} p_{ij}(a) x_{ia} = p_j \qquad$ for all $j \in S$

$\quad x_{ia} \geq 0 \qquad$ for all $i \in S, a \in A$

Observability constraints are introduced into this model using a new variable, α_{ka}, which represents the probability that action *a* is chosen for set *k*. Redefining α_{ka} in terms of the existing decision variable for the linear programming model, one obtains the equation below.

$$\alpha_{ka} = \frac{x_{ia}}{\sum_{a=1}^{M} x_{ia}} = \frac{x_{ia}}{x_i} \text{ for all } i \in S_k, a \in A \qquad (2.3)$$

This allows the original LP formulation to be rewritten in terms of the new variable, α_{ka}, and the existing decision variable, x_i, which results in a non-linear programming problem.

$$\text{Minimize} \quad \sum_{i=1}^{N} \sum_{a=1}^{M} c_{ia} \alpha_{ka} x_i$$

$$\text{Subject to} \quad \sum_{a=1}^{M} x_{ja} - \gamma \sum_{i=1}^{N} \sum_{a=1}^{M} p_{ij}(a) \alpha_{ka} x_i = p_j \qquad \text{for all } j \in S$$

$$\sum_{a=1}^{M} \alpha_{ka} = 1 \qquad \text{for all } k \in O$$

$$\alpha_{ka} \geq 0 \qquad \text{for all } k \in O, a \in A$$

$$x_i \geq 0 \qquad \text{for all } i \in S, a \in A$$

This can be expressed in vector notation as

$$\text{Minimize} \quad \Phi(\alpha) = xc(\alpha) \tag{2.4}$$

$$\text{Subject to} \quad x[I - \gamma P(\alpha)] = p \tag{2.5}$$

$$\sum_{a=1}^{M} \alpha_{ka} = 1 \qquad \text{for all } k \in O$$

$$\alpha_{ka} \geq 0 \qquad \text{for all } k \in O, a \in A$$

$$x_i \geq 0 \qquad \text{for all } i \in S, a \in A$$

The algorithm starts with an initial implementable policy. The policy is evaluated by inverting the matrix in (2.5) to find *x* and then the cost associated with policy α ($\Phi(\alpha) = xc(\alpha)$) is computed. If possible, a new policy, $\alpha`$, is found such that $\Phi(\alpha`) < \Phi(\alpha)$. A new policy is determined by using the method of feasible directions described in Bazarra and Shetty (1979). $\alpha` = (\alpha + \theta\beta^*)$ represents the new policy where β^* is the steepest descent direction and θ the stepsize. The stepsize is determined by using a search procedure to minimize the Taylor's polynomial approximation of $\Phi(\alpha + \theta\beta^*)$. The steepest descent direction is a vector $\beta = (\beta_{11}, ...\beta_{1m}, ...\beta_{K1}, ...\beta_{KM})$ that will have one component β_{k^*a} equal to1/2, one component $\beta_{k^*a'}$ equal to -1/2 and all other components set to zero. The non-zero components of the vector correspond to two

actions within one observation *k*, whose current probabilistic values α_{ka} will be increased or decreased.

In determining a new policy, a policy change is only made in one observation set and between two actions. The algorithm can be modified to allow policy changes in all sets, S_k simultaneously, but Serin and Kulkarni (1995) note this will result in slow convergence, and results are not reported using that method. In addition, the policies determined are randomized locally optimal policies.

2.2.3.3 Finite Horizon Discounted Total Cost

Serin and Avsar (1997) study the restricted observation problem for the finite horizon discounted total cost problem. Their algorithm is based on a nonlinear programming formulation of the finite horizon problem with observability constraints added. The method of feasible directions is used to solve an instance of this problem. With the nonlinear programming formulation, the authors show the feasible set for the finite horizon restricted observation problem is a polyhedral set with extreme points corresponding to deterministic policies. Therefore, a global optimal deterministic policy exists and can be found by their solution method.

$$Maximize \sum_{i=1}^{N} p_i v_{iT}$$

$$subject\ to$$

$$v_{it} \le c_{ia} + \gamma \sum_{j=1}^{N} p_{ij}(a) v_{j(t-1)} \quad \forall i \in S, a \in A, t = 1...T$$

$$v_{i0} = 0 \quad \forall i \in S$$

The above formulation is the linear programming model for the finite horizon MDP. This is converted to a finite horizon restricted observation MDP, by introducing a new decision variable, α_{kat}, which is the probability that action *a* is taken for observation set *k* at time period *t*. Rewriting the above formulation in terms of the new decision variable, yields the following (for the minimization problem)

$$Minimize\ \phi(\alpha) = \sum_{i=1}^{N} p_i v_{iT}$$

subject to

$$v_{it} \le c_{it}(\alpha) + \gamma \sum_{j=1}^{N} p_{ij}(\alpha, t) v_{j(t-1)} \quad \forall i \in S, t = 1...T$$

$$\sum_{a=1}^{M} \alpha_{kat} = 1 \quad \forall k \in O, t = 1,2..T$$

$$\alpha_{kat} \ge 0 \quad \forall a, k, t$$

$$v_{i0} = 0 \quad \forall i \in S$$

Recursive substitution of v_{it} results in the following NLP.

$$Minimize\ \phi(\alpha) = \sum_{i=1}^{N} p_i \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma^{T-t} \left(P(\alpha, T+1) P(\alpha, T) P(\alpha, T-1) ... P(\alpha, t+1) \right)_{ij} c_{jt}(\alpha)$$

subject to

$$\sum_{a=1}^{M} \alpha_{kat} = 1 \quad \forall k \in O, t = 1,2...T$$

$$\alpha_{kat} \ge 0 \ \forall a \in A, k \in O, t = 1,2...T$$

The algorithm closely mirrors the dynamic programming formulation of Howard. It begins with an initial policy, which is evaluated by calculating the expected discounted cost. Policy improvement is obtained by determining a direction that leads to a new policy with smaller cost. If so, only one alternative (for one observation set) is changed in this policy. The sequence of policy improvement and policy evaluation continues until no improving directions exist. Termination occurs at the optimal deterministic policy. The algorithm iterates over deterministic policies and guarantees a decrease in the objective value each time. For each period *t* and observation set *k* where the optimal action is *b*, the resulting optimal policy satisfies the equation below.

$$Minimum_{b \in A} w_{it} \left\{ c_{ib} + \gamma \sum_{j=1}^{N} p_{ij}(b) v_{j(t-1)} \right\} \qquad (2.6)$$

This equation depends upon relative values ($v_{j(t-1)}$) calculated from policies of the future periods and the weighted sum of the discounted probabilities (w_{it}), which is determined from the policies of the past periods. This closely resembles how policies are selected in Howard's algorithm and in the algorithm presented in chapter 3.

2.2.3.4 Infinite Horizon Average Cost

Smith (1971) developed an enumerative approach for the undiscounted infinite horizon Markov decision problem. The algorithm iterates among admissible deterministic policies (implementable policies) until an optimal policy can be determined. Smith proved that the gain associated with an admissible policy β is better than the current admissible policy α ($g^{\beta} > g^{\alpha}$) for a maximization problem if the difference in the test quantities ($d_i(\beta,\alpha)$) for the alternatives under policies α and β contain at least one recurrent state *i* where $d_i(\beta,\alpha) > 0$ and for all other states $d_i(\beta,\alpha) \geq 0$.

The test quantity is the well-known policy evaluation quantity used in Howard's dynamic programming approach (1960). In some cases, the test quantity may evaluate to a positive or negative quantity within a given state set. When this occurs, it is not possible to determine if the policy β is better. Therefore, policy β is referred to as an undetermined policy. A new iteration process is performed until the policy converges or all undetermined policies are resolved. The iteration equation is defined as

$$d^{n+1}(\beta,\alpha) = P^{\beta} d^{n}(\beta,\alpha) = P^{\beta}\left[\left(P^{\beta}\right)^{n} d(\beta,\alpha)\right]$$

Successive powers of the *P* matrix are calculated and multiplied by the decision vector, basically obtaining values for the steady state information vector, π, for each undetermined policy. Convergence is guaranteed if the iteration process is transformed into

$$d^{n+1}(\beta,\alpha) = [sP^{\beta} + (1-s)I]d^{n}(\beta,\alpha) \;\; 0 < s < 1 \; .$$

From within this process, if for any *n*, $d_i^n(\beta,\alpha) > 0$ for at least one *i* and *i* is recurrent, and $d_j^n(\beta,\alpha) \geq 0$ for all *j*, then $g^{\beta} > g^{\alpha}$. The steps defined above repeat until an optimal policy is found or all undetermined policies have been resolved. Smith notes that for very large problems, this method could degenerate into successive elimination of undetermined policies. In addition, at the enumeration stage, it is hard to keep track of which policies have been examined. An alternative evaluation technique was proposed using the entry derived semi-Markov process. The entry-derived process can reduce the dimension of the problem and make it more unlikely that undetermined policies will arise.

Hordijk and Loeve (1994) present a search heuristic that finds locally optimal solutions for the infinite horizon MDP with average cost criterion. They use the method

of successive approximations to find the locally optimal deterministic policy for the state set. Their algorithm is applied to both periodic policies and stationary limiting policies. They define the current iteration step, *n*, equal to 1. The algorithm begins with generating an initial policy (α^1), choosing an initial Π vector (x^1) and relative values (v^1). The gain $g^n(s,a)$ associated with observation set *s* under action *a* is computed as

$$\sum_{i \in S_k} x_i^n \left\{ c_{ia} + \sum_{j=1}^{N} p_{ij}(a) v_j(n) \right\}$$

The new policy α^{n+1} is the one that minimizes the cost $g^n(s,a)$ for all observations S_i. The algorithm continues by updating the estimates of the steady state pi vector and relative values using equations

$$x^{n+1} = x^n P(a^{n+1})$$

$$v^{n+1} = (a^{n+1}) + P(a^{n+1}) v^n$$

After the values have been updated, return to policy evaluation until the termination criteria ($a^{m+L} = \alpha^m$, $\left\| x^{m+L} - x^m \right\| < \varepsilon$, and $\text{span}(v^{m+L} - v^m) < \varepsilon$) has been met.

Termination occurs when the difference between the Π vectors (either for the periodic policy or stationary policy) is less than some chosen value, ε, and the policy has converged.

Hastings and Sadjadi (1979) developed a bounded enumeration algorithm applicable to solving policy constrained Markov decision problems. Their algorithm begins with solving the unconstrained problem using value iteration. Using the optimal action from the unconstrained problem, they develop bounds for the constrained problem via action ranking. Bounds on the gain of any policy can be generated from the action ranking and used to develop a ranking for the set of all admissible policies. The bounded enumeration technique applies value iteration to the admissible policies in the ranked set until the gain of the current best policy is greater than the upper bound of the policy being evaluated. An action difference ($d(n,i,k)$) for each state and possible action is determined from the expected *n* and *n*-1 stage rewards as follows:

$$d(n,i,k) = v_{i,n}(k) - v_{i,n-1}$$

$$v_{in}(k) = c_{ia} + \sum_{j=1}^{N} p_{ij}(k) v_{j,n-1}$$

$$v_{j,n-1} = \max_k \left(c_{ia} + \sum_{j=1}^{N} p_{ij}(k) v_{j,n-2} \right)$$

The optimal action for the unconstrained problem as *n* tends to infinity can be determined and used as the *n*-1 stage reward in the equations above. The action differences are calculated for all possible states and actions. The admissible policies are then ranked by determining the upper and lower bounds on the gain using the action differences. Given a policy α, the upper bound $U(\alpha) = \max_k (d(n,i,k))$ over all states *i* in the policy. Similarly, the lower bound $L(\alpha) = \min_k (d(n,i,k))$ over all states in the policy. The bounded enumeration is performed on the ranked policy list. The algorithm iterates through each policy, performing value iteration and updating the bounds and best gain. For large problems, the authors suggest stopping at the best policy obtained after some number of iterations, or some epsilon optimal policy.

2.2.4 Applicability of previous work to current problem

The model of no information sharing is an MDP with Restricted Observations. We use the completely observable Markov model to define a partition of states where all states in the partition must have the same control decision. Conceptually, this can be thought of as a partially observable system where the observable outputs are some aspect of the state space. For example, the observable output for a first stage decision maker in a two-stage supply chain, with state space defined as inventory level at each stage, would be his/her own inventory position. While the completely observable model reflects information that is known at both stages, we want to quantify the difference associated with the level of information sharing by comparing the optimal control decision and the gain of the process under all levels. In the POMDP, one must find the optimal control for the entire space of vectors, which is not easily implementable or usable. The restricted observation model determines the optimal control alternative the decision maker would choose given he possesses partial information about the process; his observable outputs.

Therefore, we are seeking the optimal control decision for the available outputs that are linked to the internal process. The optimal gain and associated control policy when information is shared can be determined using Howard's (1960) dynamic programming solution method. The algorithm proposed in this research, can determine the optimal gain and associated control policy for the undiscounted average cost case, when no information is shared. The algorithm determines the policy by altering Howard's (1960) policy iteration to calculate the expected transition cost as a function of the payoff structure for each state in the set and their corresponding steady state probabilities.

Existing research for finding optimal policies for the restricted observation undiscounted cost case have been dominated by enumerative based searches. Although these have provided acceptable locally optimum solutions for their applications, a more robust and computationally efficient algorithm must be used to study the information sharing application. In addition, some of the heuristics have no guaranteed performance bound (Hordijk and Loeve, 1994 and Smith, 1971) or guaranteed deterministic policy (Serin and Avsar, 1997). The method proposed by Smith (1971), and Hastings and Sadjadi (1979) cannot handle very large problems, as their algorithmic structure is enumeration based. The algorithm by Hordijk and Loeve (1994) finds locally optimal solutions. However, the best performance that algorithm could achieve for the information sharing problem, if the information vector and the relative values were solved for explicitly, is at most 54% of the problems solved optimally and the remaining with an average relative error of approximately 6%. Our algorithm without randomization serves as an upper bound on the possible solution obtained from their algorithm. The details of the algorithm are discussed in chapter 3.

Chapter 3 Heuristic for MDPs with Restricted observations

3.1. Background

In this chapter, we present a computationally efficient procedure to determine control policies for an infinite horizon Markov Decision process with restricted observations (ROMDP). The optimal policy for the system with restricted observations is a function of the observation process and not the unobservable states of the system. Thus, the policy is stationary with respect to the partitioned state space. Recall for a partially observable MDP, the optimal policies are not stationary with respect to the original process but are functions of all possible states of the information vector, also known as a 'belief-state'.

Serin and Kulkarni (1995) develop an algorithm that finds locally optimal policies for the infinite horizon discounted cost case. The algorithm we propose addresses the undiscounted average cost case. Algorithms for the infinite horizon undiscounted cost problem are presented by Smith (1971), Hordijk and Loeve (1994), and Hastings and Sadjadi (1979). The algorithms developed by Hastings and Sadjadi (1979) are enumerative based and thus intractable for large problems. The algorithm developed by Smith (1979) is a policy iteration algorithm containing an enumerative component that is used when a better policy can not be determined. All of the algorithms developed to address the infinite horizon average reward ROMDP provide local optimal policies. In contrast, the algorithm we present combines a local search with a modified version of Howard's (1960) policy iteration method to provide optimal or near-optimal policies. We demonstrate empirically that the algorithm finds the optimal deterministic policy for over 97% of the problem instances generated. In the instances where the optimal policy cannot be determined, the average error is close to one percent.

3.2. An algorithm for the undiscounted case

3.2.1. Background and notation

The process being analyzed is a Markov Decision Process with state space S and action space A. The state of the system can not be observed, however some output of the system is observable. Based on those outputs, one can infer the state or possible states the system may be in. We seek to find an optimal control policy defined over the observation process that

minimizes the long term average cost. The optimal policy has the property that each state within a given observation set takes the same action. Let O represent the set of observable outputs that partitions the state space into mutually exclusive sets S_k. Each state $i \in S$ is a member of only one observable set defined by a mapping function G_i. A summary of the problem notation is presented below.

A: The set of available actions $\{1 \ldots M\}$

S: The set of possible states $\{1 \ldots N\}$.

O: The set of observable outputs $\{1 \ldots K\}$.

$G(i)$: A function mapping a state i to a single observable output in the set O.

S_k: A given partition of the state space S satisfying $\{i: G(i) = k\}$.

$A(k)$: The set of admissible actions for observation set S_k. $A(k) \subseteq A$.

X_n: A random variable denoting the state of the system at time n=0,1…

A_n: The action chosen at time n.

$p_{ij}(a)$: The one step transition probability from state i to j under alternative $a \in A$.

$$p_{ij}(a) = P\{X_{n+1} = j \mid X_n = i, A_n = a\}$$

c_{ia} : The immediate reward associated with transitioning to state i under alternative $a \in A$. $c_{ia} = E\{C(X_n, A_n) \mid X_n = i, A_n = a\}$. In Howard's (1960) policy iteration algorithm, this quantity is denoted q_i^a .

3.2.2 Model and Solution Method for ROMDP

3.2.2.1 Mathematical Model for Infinite Horizon average cost ROMDP

Wolf and Dantzig (1962) develop a linear programming formulation for the infinite horizon completely observable MDP under the average cost criterion. The model is briefly described below.

$$\min \sum_a \sum_i c_{ia} x_{ia}$$

subject to

$$\sum_a x_{ia} = \sum_a \sum_j x_{ia} p_{ji}^a \ \forall i \in S$$

$$\sum_a \sum_i x_{ia} = 1$$

$$x_{ia} \geq 0 \ \forall i \in S, a \in A$$

The decision variable x_{ia} denotes the long run proportion of time the system is in state *i* under alternative *a*. Wolf and Dantzig (1962) show the optimal policy will always be deterministic. That is only one alternative *a* will be selected for each state such that $x_{ia} > 0$. The dual formulation to the above linear programming problem is

max g

subject to

$$g + v_i - \sum_j p_{ij}^a v_j \le c_{ia} \ \forall i \in S, a \in A$$

v_j free $\forall j \in S$.

The infinite horizon average cost case can be formulated as an MDP with restricted observations by adding constraints to the primal model of Wolf and Dantzig (1962) to represent the observability restrictions of the process. Let α_{ka} denote the probability of choosing action *a* for observation set *k*. The probability of choosing action *a* for any state *i* is defined as

$$\frac{x_{ia}}{\sum_a x_{ia}}.$$

In order for all states *i* in observation set S_k to take the same action at a point in time, the following must hold for every observation set S_k.

$$\frac{x_{ia}}{\sum_a x_{ia}} = \alpha_{ka} \ \forall i \in S_k, a \in A.$$

This constraint ensures all states have the same probability of choosing action *k*. In addition to this constraint, only one action can be taken for any given observation set S_k. This ensures the resulting policy is deterministic.

$$\sum_a \alpha_{ka} = 1$$

$$\alpha_{ka} \in \{0,1\}$$

Let $x_i = \sum_a x_{ia}$ then, $x_{ia} = \alpha_{ka} \sum_a x_{ia} = \alpha_{ka} x_i$. The primal LP with the observability constraints added is shown below.

$$\min \sum_{i,a} c_{ia} x_i \alpha_{G(i)a}$$

subject to

$$x_i = \sum_{a,j} x_j \alpha_{G(j)a} p_{ji}(a) \; \forall i \in S$$

$$\sum_i x_i = 1$$

$$\sum_a \alpha_{ka} = 1 \quad \forall k \in O$$

$$x_i \geq 0 \; \forall i \in S$$

$$\alpha_{ka} \in \{0,1\} \; \forall k \in O, a \in A$$

Let $c_i(\alpha) = \sum_a c_{ia} \alpha_{ka}$ $\mathbf{c}(a) = [c_1(\alpha)\; c_2(\alpha) ... c_N(\alpha)]$. Here, $c_i(\boldsymbol{\alpha})$ is immediate reward associated with state *i* under policy α. Let $\mathbf{P}(\boldsymbol{\alpha})$ be the matrix defined with entries $p_{ij}(\boldsymbol{\alpha})$ where $p_{ij}(\alpha) = \sum_a \alpha_{G(i)a} p_{ij}(a)$.

Then the *NLP* can be written in matrix notation as

$$\min \Phi(\alpha) = \bar{x}\mathbf{c}(\alpha)$$

subject to

$$\bar{x}[I - \mathbf{P}(\alpha)] = 0 \qquad (3.1)$$

$$\sum_i x_i = 1 \qquad (3.2)$$

$$\sum_a \alpha_{ka} = 1 \quad \forall k \in O \qquad (3.3)$$

$$x_i \geq 0 \; \forall i \in S \qquad (3.4)$$

$$\alpha_{ka} \in \{0,1\} \; \forall k \in O, a \in A \qquad (3.5)$$

The above problem is a mixed integer nonlinear programming problem (MINLP). There is no exact solution known for solving this problem and thus heuristic methods must be used. We exploit some mathematical properties of the above stated problem to provide insight into developing a simple heuristic. . Therefore, we relax the integrality constraints on α_{ka} to derive the gradient vector with respect to α_{ka}. The gradient will be used as a guide to finding a better deterministic policy.

3.2.2.2 Characterization of Feasible Descent Direction

We first derive the gradient of the objective function at the point α ($\nabla\Phi(\alpha)$).

$$\frac{\partial\Phi(\alpha)}{\partial\alpha_{ka}} = \sum_i \frac{\partial c_i(\alpha)}{\partial\alpha_{ka}} x_i + \sum_i \frac{\partial x_i}{\partial\alpha_{ka}} c_i(\alpha)$$

Since,

$$\frac{\partial c_i(\alpha)}{\partial\alpha_{ka}} = \frac{\partial}{\partial\alpha_{ka}}\left(\sum_a c_{ia}\alpha_{G(i)a}\right) = \begin{Bmatrix} c_{ia} & if\ G(i)=k \\ 0 & otherwise \end{Bmatrix}$$

we have,

$$\frac{\partial\Phi(\alpha)}{\partial\alpha_{ka}} = \sum_{i\in S_k} c_{ia} x_i + \sum_i \frac{\partial x_i}{\partial\alpha_{ka}} c_i(\alpha).$$

Let $\frac{\partial\bar{x}}{\partial\alpha_{ka}} = \left[\frac{\partial x_1}{\partial\alpha_{ka}}, \frac{\partial x_2}{\partial\alpha_{ka}} \ldots \frac{\partial x_N}{\partial\alpha_{ka}}\right]$ and let $\mathbf{P}^{ka}$ be the matrix defined with entries

$$p_{ij}^{ka} = \begin{Bmatrix} p_{ij}(a) & if\ i\in S_k \\ 0 & Otherwise \end{Bmatrix}.$$

Then from constraint 3.1 above

$$\frac{\partial\bar{x}}{\partial\alpha_{ka}}\left[\mathbf{I} - \mathbf{P}(\alpha)\right] = \bar{x}\mathbf{P}^{ka}.$$

The matrix [**I**-**P**(α)] is not invertible because it contains a redundant constraint. However, we make it invertible by arbitrarily replacing the N^{th} constraint with the following equation.

$$\sum_i \frac{\partial x_i}{\partial\alpha_{ka}} = 0$$

This equation represents the partial derivative with respect to α_{ka} for constraint 3.2. Replace the N^{th} column of the matrix [**I**-**P**(α)] with all ones and let this transformed matrix be defined as

$$\mathbf{Q}(\alpha) = \begin{bmatrix} 1-p_{11}(\alpha) & -p_{12}(\alpha) & \cdots & -p_{1,N-1}(\alpha) & 1 \\ -p_{21}(\alpha) & 1-p_{22}(\alpha) & \cdots & -p_{2,N-1}(\alpha) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -p_{N1}(\alpha) & -p_{N2}(\alpha) & \cdots & -p_{N,N-1} & 1 \end{bmatrix}$$

Replace the N^{th} column of the matrix $\mathbf{P}^{ka}$ with all zeros. As a result of this transformation, [**Q**(α)]$^{-1}$ exists and

$$\frac{\partial\bar{x}}{\partial\alpha_{ka}} = \bar{x}\mathbf{P}^{ka}\left[\mathbf{Q}(\alpha)\right]^{-1}.$$

Then

$$\frac{\partial\Phi(\alpha)}{\partial\alpha_{ka}} = \sum_{i\in S_k} c_{ia}x_i + \bar{x}\mathbf{P}^{ka}\left[\mathbf{Q}(\alpha)\right]^{-1}\mathbf{c}(\alpha)$$

Let $\bar{v} = [v_1 ... v_n]$ be the solution to $\mathbf{Q}(\boldsymbol{\alpha})\bar{v} = \mathbf{c}(\boldsymbol{\alpha})$. Then we can write the gradient of the objective function with respect to a given policy as follows as

$$\nabla\Phi(\alpha) = \left[\frac{\partial\Phi(\alpha)}{\partial\alpha_{11}}\cdots\frac{\partial\Phi(\alpha)}{\partial\alpha_{1M}}, \frac{\partial\Phi(\alpha)}{\partial\alpha_{21}}\cdots\frac{\partial\Phi(\alpha)}{\partial\alpha_{KM}}\right]$$

where,

$$\frac{\partial\Phi(\alpha)}{\partial\alpha_{ka}} = \sum_{i\in S_k} x_i\left(c_{ia} + \sum_j p_{ij}(\alpha)v_j\right)$$

Note, this equation is simply the policy improvement test statistic for a given observation set S_k. The solution to $\bar{v} = [\mathbf{Q}(\boldsymbol{\alpha})]^{-1}\mathbf{c}(\boldsymbol{\alpha})$ is in fact the relative values obtained from solving the simultaneous equations from the Howard (1960) value determination step with v_N set to 0. With $v_N = 0$, we have $[\mathbf{I}\text{-}\mathbf{P}(\boldsymbol{\alpha})]\bar{v} = \mathbf{c}(\boldsymbol{\alpha})$ which is equivalent to

$$v_i - \sum_j p_{ij}(\alpha)v_j + g = c_i(\alpha) \Rightarrow v_i + g = c_i(\alpha) + \sum_j p_{ij}v_j \,.$$

Although the matrix $\mathbf{P}^{ka}$ was modified above to remove the redundant constraint, the modification corresponds to the column where $v_N = 0$ so that

$$\bar{x}\mathbf{P}^{ka} = \sum_{i\in S_k} x_i \sum_{j=1}^{N-1} p_{ij}v_j = \sum_{i\in S_k} x_i \sum_{j=1}^{N} p_{ij}v_j \,.$$

So, we obtain both $\bar{x} = \mathbf{b}\,[\mathbf{Q}(\boldsymbol{\alpha})]^{-1}$ and $\bar{v} = [\mathbf{Q}(\boldsymbol{\alpha})]^{-1}\mathbf{c}(\boldsymbol{\alpha})$, with $\bar{v} = [v_1 \ldots v_{N-1}, g]$, g representing the gain of the process ($\Phi(\alpha)$) and $\mathbf{b}$ an N-element vector defined as [0,...,0,1]. Using the definition of $\mathbf{Q}(\boldsymbol{\alpha})$ and $\mathbf{b}$ above, the NLP can be rewritten as

$$\min \Phi(\alpha) = \bar{x}\mathbf{c}(\alpha)$$
subject to
$$\bar{x}\mathbf{Q}(\alpha) = \mathbf{b}$$
$$x_i \geq 0 \;\forall i \in S$$
$$\alpha_{ka} \in \{0,1\} \;\forall k \in O, a \in A$$

As a result of this transformation and the existence of $[\mathbf{Q}(\boldsymbol{\alpha})]^{-1}$ for any policy $\boldsymbol{\alpha}$, there corresponds a unique vector $\bar{x}$ such that $(\boldsymbol{\alpha}, \bar{x})$ is a feasible solution to the problem. So the

policy variable $\boldsymbol{\alpha}$ can be considered the only variable of the model. The problem can be equivalently stated as

$$\min \Phi(\alpha) = \mathbf{b}\left[\mathbf{Q}(\alpha)\right]^{-1}\mathbf{c}(\alpha)$$

subject to

$$\sum_a \alpha_{ka} = 1 \quad \forall k \in O$$

$$\alpha_{ka} \in \{0,1\}\, \forall k \in O, a \in A$$

If the integrality constraints on the policy variable are relaxed, and an instance of this problem is solved using the method of feasible directions, then we must find a feasible direction, $\boldsymbol{\beta} = \left[\beta_{11,}\beta_{12},...\beta_{1M},...\beta_{KM}\right]$ such that $\boldsymbol{\alpha} + \theta\boldsymbol{\beta} \in A$ and $\nabla\Phi(\alpha)^{t}\,\boldsymbol{\beta} < 0$ for some $\theta > 0$. If this direction can be found, then the maximum distance, θ, which can be traveled along that direction such that $\Phi(\boldsymbol{\alpha} + \theta\boldsymbol{\beta})$ is minimized must be determined. First, the gradient direction will be discussed followed by the computation of the maximum distance (step size) of θ. Since we are only concerned with deterministic policies, the gradient direction will be used as a guide to finding a better deterministic policy. As derived above, the gradient direction reduces to the policy improvement test statistic of Howard(1960) weighted by the state probabilities associated with a given observation set. From Serin(1989), a feasible direction $\boldsymbol{\beta}$ associated with an improving policy, satisfies

$$\begin{aligned} \beta_{ka} &\geq 0 \ \textit{for} \quad \alpha_{ka} = 0 \\ \beta_{ka} &\leq 0 \ \textit{for} \quad \alpha_{ka} = 1 \end{aligned}.$$

For a given observation set S_k, let $U_k(\boldsymbol{\alpha}) = \{a \in A(k) |\ \alpha_{ka} < 1\}$ and $u(k)$ represent an alternative for set S_k that minimizes the directional derivative. That is, select $u(k)$ such that

$$u(k) = \min_{a \in A(k)} \frac{\partial\Phi(\alpha)}{\partial\alpha_{ka}}.$$

Further, let $d(k)$ represent the current alternative a. Then it follows that $u(k)$ is an improving alternative if

$$\frac{\partial\Phi(\alpha)}{\alpha_{ku(k)}} < \frac{\partial\Phi(\alpha)}{\alpha_{kd(k)}} \Rightarrow$$

$$r(k) = \left(\frac{\partial\Phi(\alpha)}{\alpha_{ku(k)}} - \frac{\partial\Phi(\alpha)}{a_{ka}}\right) < 0.$$

A value $r(k)$ can be defined for all S_k, $k \in O$. Since α_{ka} can only take on values in the discrete set {0,1}, then to move in a direction that minimizes the directional derivative, β_{ka} can only take on values {1,0,-1}. Formally,

$$\beta_{ka} = \begin{cases} 1 & if\ a = u(k) \\ -1 & if\ a = d(k) \\ 0 & otherwise \end{cases}$$

which implies the new policy α' is defined by

$$\alpha'_{ka} = \alpha_{ka} + \theta\beta_{ka}.$$

Since we are only concerned with feasible policies that are deterministic, then the only feasible (and maximal) value that θ can take is one. Any value less than one would lead to a randomized and thus infeasible policy for this problem. If the optimal value of θ that minimizes $\Phi(\boldsymbol{\alpha}+\theta\boldsymbol{\beta})$ is less than one, than the new policy $\boldsymbol{\alpha}+\theta\boldsymbol{\beta}$ will result in an objective function value which is worse. As illustrated in Serin and Kulkarni (1995), an approximation of the optimal θ can be determined using Taylor's polynomial approximation. For the purposes of this algorithm, this evaluation is not needed. It is sufficient to simply move in the direction of β at the maximal value and compute the resulting gain. If the gain is better, find a new direction to travel at this new policy α'. If the gain is worse, then stop at the current policy α that has been found.

3.2.2.3 Policy Iteration Heuristic for ROMDP

To solve the problem, it is not necessary to construct a *K*M* vector in terms of the decision variable $\boldsymbol{\alpha}$ to represent an implementable policy. It is sufficient to only carry the information needed in terms of the action taken for a given observation set S_k. Therefore, let $\pi = [\pi_1..\pi_K]$ represent the implementable policy vector for the observed process, defined in terms of the action space *A*. Then

$$\pi_k = a\ if\ \alpha_{ka} = 1.$$

Then $\mathbf{Q}(\boldsymbol{\alpha})$ has entries $p_{ij}(\boldsymbol{\alpha})$ where

$$p_{ij}(\alpha) = \sum_k \alpha_{G(i),k}\, p_{ij}(a) = p_{ij}\left(\pi_{G(i)}\right).$$

Similarly, the vector $\mathbf{c}(\boldsymbol{\alpha})$ has entries $c_i(\boldsymbol{\alpha})$ where

$$c_i(\alpha) = \sum_j c_{ia}\alpha_{ka} = c_{i,\pi_{G(i)}} .$$

These substitutions are denoted as $\mathbf{Q}(\boldsymbol{\alpha}\,|\boldsymbol{\pi})$ and $\mathbf{c}(\boldsymbol{\alpha}\,|\boldsymbol{\pi})$. The heuristic for finding an admissible policy is summarized below. The cost $\Phi(\boldsymbol{\alpha})$ associated with a feasible policy $\boldsymbol{\alpha}$ will hereafter be denoted by the gain *g*.

Step 0. Initialization

Generate an initial admissible policy π.

Set $g^* = \infty$.

Step 1.Policy Evaluation

Determine the gain (g^{π}), relative values (v_i) and steady state probabilities $\bar{x}$ associated with policy $\boldsymbol{\pi}$ using $\bar{x} = \mathbf{b}[\mathbf{Q}(\boldsymbol{\alpha}\,|\boldsymbol{\pi})]^{-1}$ and $\bar{v} = [\mathbf{Q}(\boldsymbol{\alpha}\,|\boldsymbol{\pi})]^{-1}\mathbf{c}(\alpha \mid \pi)$.

(*a*). If $g^{\pi} < g^*$, set $g^* = g^{\pi}$ and proceed to Step 2.

(*b*) If $g^{\pi} \geq g^*$, the current solution g^* is a local minimum.

Step 2. Policy Improvement.

For all $k \in O$ find an action π_k that minimizes the directional derivative

$$\pi_k = \min_{a \in A(k)} \sum_{i \in S_k} x_i (c_{ia} + \sum_{j=1}^{N} p_{ij}(a) v_j) .$$

In the algorithm proposed by Serin and Kulkarni (1995) for the infinite horizon discounted total cost case, only one observation set (k^*) is chosen during the policy improvement step corresponding to

$$k^* = \arg\min_{k \in O} r(k) .$$

The new alternative corresponds to the steepest descent direction. For the purposes of this heuristic, any $k \in O$ satisfying $r(k) < 0$ will be chosen to construct a new policy. Although the resulting direction may not be the steepest descent direction, it will still satisfy $\nabla\Phi(\alpha)^t\boldsymbol{\beta} < 0$ and can possibly result in a smaller objective function value.

Lemma 3.1: Let $\boldsymbol{\alpha} = [\alpha_{11},..\alpha_{KM}]$ be a solution to an instance of the ROMDP.
$\boldsymbol{\alpha}$ is a local minimum if either one of the following properties is true:

1. No feasible direction $\boldsymbol{\beta}$ can be found such that $\nabla\Phi(\boldsymbol{\alpha})^t\boldsymbol{\beta} < 0$

2. A feasible direction β was determined using policy improvement, but $\Phi(\alpha+\theta\beta) > \Phi(\alpha)$

If condition 1 is met, α is referred to as a type 1 local optimal solution. If condition 2 is met, α is referred to as a type 2 local optimal solution.

Proof: Assume α is not a local optimal solution. This implies there exists some $k \in O$ and some alternative $b \in A(k)$ such that

$$\sum_{i \in S_K} x_i \left(q_i^b + \sum_j p_{ij}^b v_j \right) < \sum_{i \in S_k} x_i \left(q_i^{\alpha_k} + \sum_j p_{ij}^{\alpha_k} v_j \right) \Rightarrow$$

$$\frac{\partial \Phi(\alpha)}{\partial \alpha_{kb}} < \frac{\partial \Phi(\alpha)}{\alpha_{k\alpha_k}} \Rightarrow r(k) < 0$$

This implies b is an alternative that minimizes the directional derivative and a feasible direction β exists such that $\sum_k r(k) = \nabla\Phi(\alpha)^t \beta < 0$. If no feasible direction is found, then the previous condition is not satisfied and the current solution is a local optimal solution. Assume a feasible direction exists such that $\nabla\Phi(\alpha)^t\beta < 0$ and the optimal value of θ that minimizes $\Phi(\alpha+\theta\beta)$ is less than one. If the new policy $\alpha`$ is constructed such that $\alpha` = \alpha+\beta$, then $\Phi(\alpha+\beta) > \Phi(\alpha)$. This implies the step size assumption of $\theta = 1$ resulted in a move along the feasible direction which was too large, and thus a smaller objective function value was not attained. Therefore, the current solution α is a local minimum (or critical point) of the current problem. Q.E.D.

Lemma 3.2: As long as there exists at least one local optimal solution, the algorithm defined above will terminate after a finite number of iterations.

Proof: Let L: = {α | Lemma 3.1 satisfied}. It follows for any $\bar{\alpha} \in \bar{L}$, there always exists a new policy δ found by policy improvement such that $\Phi(\delta) < \Phi(\bar{\alpha})$. If $\delta \in \bar{L}$, policy improvement continues. If $\delta \in L$, then one further iteration of policy improvement is performed and algorithm terminates.

Since $\bar{L}$ is a finite set of policies which always contain a feasible direction yielding a better objective function value, it is impossible for policies within $\bar{L}$ to cycle amongst each other. Therefore, using the gradient vector to find a better policy is guaranteed to terminate on a local minimum. Q.E.D.

When the heuristic terminates on a local minimum characterized by Lemma 3.1, this convergence will be referred to as policy iteration convergence. This minimum is not guaranteed to be the global minimum unless $|L| = 1$. Therefore, the policy iteration heuristic for the ROMDP is augmented by a local improvement procedure to increase the probability of finding the global minimum.

3.2.2.4 Local Improvement Procedure

Solution representation

A solution is represented as an ordered sequence of alternatives denoting the actions taken for each observation set S_k. Let $\boldsymbol{\pi}$ denote the policy vector of alternatives.

Neighborhood Structure

For a given policy $\boldsymbol{\pi}$ and a candidate solution $\bar{\pi}$, new neighbor $\boldsymbol{\pi}$` is constructed by interchanging elements of $\boldsymbol{\pi}$ with $\bar{\pi}$. The number of elements to be interchanged depends on the type of neighborhood construction algorithm (policy perturbation) being performed. Policy perturbations will be described in detail in the following section. Selection of a candidate solution $\bar{\pi}$ is determined from $\boldsymbol{\pi}$ by minimizing the directional derivative at $\boldsymbol{\pi}$. This can be easily obtained from the policy improvement step of policy iteration, as the algorithm will terminate on a local minimum only after having found no improving direction. This assumes the local minimum is a type 2 local minimum solution. If the local minimum is a type 1 solution, then the policy constructed from choosing the second best alternative for all k that minimizes the directional derivative can be chosen as $\bar{\pi}$. Any neighbor constructed in this manner will be a feasible solution to the ROMDP.

Evaluation of a Neighbor

The value of the new policy π`is evaluated using the same equations defined in the policy evaluation step of policy iteration. If the current solution is an improvement over the current best solution, then the policy iteration heuristic is restarted using the new neighboring solution as a starting point.

Implementation Details

The steps for the local improvement procedure are summarized below:

Step 0. Initialization

Construct a neighbor $\boldsymbol{\pi}` \in N(\boldsymbol{\pi})$,where $\boldsymbol{\pi}$ is local minimum from policy iteration convergence.

Set $g^* = g^{\pi}$

Step 1.Policy Evaluation

(*a*) Determine the gain ($g`$), relative values (v_i) associated with policy $\boldsymbol{\pi}$ using

$\bar{v} = [\mathbf{Q}(\boldsymbol{\alpha} \mid \boldsymbol{\pi}`)]^{-1}\mathbf{c}(\boldsymbol{\alpha} \mid \boldsymbol{\pi}`)$.

(*b*) If $g` < g^*$, set $g^* = g`$ and restart policy iteration procedure.

(*c*) If $g` \geq g^*$, construct new neighbor $\boldsymbol{\pi}`$ and return to 1(*a*).

Step 2. Termination Criteria.

If $g` \geq g^*$ for all $\boldsymbol{\pi}` \in N(\boldsymbol{\pi})$, terminate with current local optimal solution.

Combining the local improvement and policy iteration procedures we have the following heuristic.

Step 0. Initialization

Generate an initial admissible policy π.

Set $g^* = \infty$.

Step 1.Policy Evaluation

(*a*) Determine the gain (g^{π}), relative values (v_i) associated with policy π using

$\bar{v} = [\mathbf{Q}(\alpha \mid \pi)]^{-1}\mathbf{c}(\alpha \mid \pi)$.

(*b*) Determine the steady state probabilities $\bar{x}$ associated with policy π using

$\bar{X} = \mathbf{b}[Q(\alpha \mid \pi)]^{-1}$

(*c*) If $g^{\pi} < g^*$, set $g^* = g$ and proceed to Step 2.

(*d*) If $g^{\pi} \geq g^*$, proceed to Step 3..

Step 2. Policy Improvement.

For all $k \in O$ find an action π_k that minimizes the directional derivative and return to step 1.

$$\pi_k = \min_{a \in A(k)} \sum_{i \in S_k} x_i \left(c_{ia} + \sum_{j=1}^{N} p_{ij}(a) v_j\right)$$

Step 3. Initialization-Local Improvement

Construct a neighbor $\boldsymbol{\pi}` \in N(\boldsymbol{\pi})$,where $\boldsymbol{\pi}$ is local minimum from policy iteration convergence.

Set $g^* = g^{\pi}$.

Step 4.Policy Evaluation

(*a*) Determine the gain ($g`$), relative values ($v_i$) associated with policy $\alpha`$ using $\bar{v} = [\mathbf{Q}(\boldsymbol{\alpha}|\boldsymbol{\pi}`)]^{-1}\mathbf{c}(\boldsymbol{\alpha}|\boldsymbol{\pi}`)$.

(*b*) If $g` < g^*$, set $g^* = g`$, $\boldsymbol{\pi} = \boldsymbol{\pi}`$and proceed to step 1*b*.

(*c*) If $g` \geq g^*$, construct new neighbor $\pi`$ and return to 4(*a*).

Step 5. Termination Criteria.

If $g` \geq g^*$ for all $\pi` \in N(\pi)$, terminate with current local optimal solution.

3.2.2.5 Neighborhood based on Policy vector

There are several ways in which a neighboring solution can be constructed. Two methods will be discussed here. The first method, referred to as policy perturbation 1 (*pp1*), takes one alternative from π and replaces it with the alternative in the same position from the candidate solution $\bar{\pi}$. Formally, for each index [*i*] , construct a neighbor $\boldsymbol{\pi}`$ such that

$$\pi`_{[i]} = \bar{\pi}_{[i]}$$
$$\pi`_{[j]} = \pi_{[j]} \quad \forall j \neq i$$

At most *K* neighbors can be generated under this neighborhood construction scheme. The second method, policy perturbation 2 (*pp2*) is similar to *pp1* with the exception that two indices are chosen instead of 1. At most $\binom{K}{2}$ neighbors can be generated under this neighborhood construction scheme.

3.2.2.6 Search Strategies and other considerations

In the above mentioned neighborhood construction schemes, as soon as an improving solution is found, we move to it and then restart the policy iteration phase of the heuristic. Alternatively, we could evaluate all neighbors of a given solution and then move to the best neighbor. This search method is also examined and is denoted as *pp1A* in the experimental

results section. This method is only examined with the neighborhood constructed using method *pp1*, since the number of neighbors to evaluate is significantly smaller than *pp2*.

Another search method deals with the selection of the candidate solution $\bar{\pi}$. Recall, in the previous section, that a candidate solution is determined by minimizing the directional derivative at the current local minimum π. From the policy improvement step, two policies can be constructed representing the first and second best alternative for each observation set S_k. Each candidate solution can serve as a starting point for the local improvement procedure, thus increasing the possible neighboring solutions evaluated. This procedure is also examined in the experimental results section and denoted *meta1*.

3.2.2.7. Neighborhood based on Information Vector

The algorithm discussed above can also be modified to consider neighborhoods based on the steady state information vector ($\bar{x}$) instead of the policy vector. If the optimal steady state vector is known, then the optimal policy associated with that vector can be determined. Therefore, better results may be achieved by randomizing the current iterate of the information vector instead of the current iterate of the policy vector. Modification of the local improvement algorithm to consider perturbations based on the steady state vector is summarized below. The policy iteration phase of the algorithm is identical and omitted here.

Step 3. Initialization-Local Improvement

Construct a neighbor $\boldsymbol{x}` \in N(\bar{x})$, where $\bar{x}$ is steady state information vector associated with the local minimum policy π from policy iterationl convergence.

Set $g^* = g^{\pi}$.

Step 4. Policy Improvement

Find a new policy based on the new information vector $x`$.

For all $k \in O$ find an action π_k that minimizes the directional derivative

$$\pi_k = \min_{a \in A(k)} \sum_{i \in S_k} x`_i (c_{ia} + \sum_{j=1}^{N} p_{ij}(a) v_j)$$

Step 5.Policy Evaluation

(*a*)Determine the gain ($g`$), relative values ($v_i$) associated with policy $\alpha`$ using

$\bar{v}$ =[**Q**(**α**|**π**`)]$^{-1}$**c**(**α**|**π**`).

(*b*). If $g` < g^*$, set $g^*$=g`, $\pi=\pi`$ and proceed to Step 1*b*.

(*c*) If $g` \geq g^*$, construct new neighbor $\boldsymbol{x}`$ and return to Step 4.

Step 6. Termination Criteria.

Terminate the algorithm after *T* total perturbations have been performed.

There are certain issues to consider when using a neighborhood based on the steady state information vector instead of the policy vector. Since the vector is a discrete set of continuous values, the perturbation method chosen can generate an infinite number of neighbors. Therefore, terminating after all neighbors have been examined (as in the policy perturbation case) is not suitable. The termination counter must be set large enough to ensure a good selection of neighbors is examined without compromising the execution time. The actual method chosen to generate neighbors is also important. Four methods are discussed below.

3.2.4.1 Perturbation methods for steady state information vector

Four perturbation methods are evaluated. Each involves modifying the value of the information vector by some step size ε. The value of ε is either randomly generated or is a constant value. Let the starting vector for generating neighbors, $\bar{x}^{\pi}$, corresponds to the steady state information vector associated with the local minimum policy π obtained from policy iteration convergence. Perturbation method 1 (*pi1*) adds a constant value, ε, to each element in the information vector. A neighbor $x`$ is constructed in the following manner with perturbation *pi1*.

Step 1.

$$x`_i = x_i^{\pi} + \varepsilon \;\forall i$$

Step 2.

$$x``_i = \frac{x`_i}{\sum_{i=1}^{N} x`_i} \quad \forall i$$

The normalization step (Step 2) is performed for all perturbation methods and will be omitted in the subsequent sections.

Method 2 (*pi2*) randomly adds and subtracts a constant value, ε, to each element in the vector and then normalizes the vector as defined in Step 2 above. A neighbor $x`$ is constructed in the following manner with perturbation *pi2*.

Step 1.

(a) Generate a random number u_i in [0,1] for state *i*.

(b) $x`_i = \max(0, x_i^{\pi} - \varepsilon)$ *if* $u < 0.5$

$x`_i = x_i^{\pi} + \varepsilon$ *if* $u \geq 0.5 \ \forall i$

With method 3 (*pi3*), a uniformly distributed number between 0 and ε is added to each element in the information vector. A neighbor $x`$ is constructed in the following manner with perturbation *pi3*.

Step 1.

i) u_i = randomly generated number between 0 and 1 for state i.

ii) $x`_i = x_i^{\pi} + u_i * \varepsilon \ \forall i$

The last method (*pi4*) performs the same type of perturbation as method 3, with the exception that the quantity added to each element in the information vector is between $\pm\varepsilon/2$. A neighbor $x`$ is constructed in the following manner with perturbation *pi4*.

Step 1.

(a) u_i = randomly generated number between 0 and 1 for state i.

(b) $x`_i = \max\left(0, x_i^{\pi} - (\varepsilon / 2) + u_i * \varepsilon\right) \forall i$

It should be noted that if a constant value of ε is chosen, then any neighbor constructed using *pi1* will always be the same. The way to avoid this is to consider two possible methods of generating neighbors. The first method always generates a different value of ε at each iteration. Therefore, the termination counter *T* defines the total number of neighbors evaluated for vector *x*. This generates neighbors more localized to *x*. An alternative approach retains the constant value of ε but chooses a new starting point during each iteration. This results in a broader search space since movement is allowed away from the current best value. In effect, only one neighbor, $x`$, is generated and evaluated. If a better objective function value is not found, then the next neighbor generated is in $N(x`)$. In this respect, the termination counter *T* defines the number of perturbations performed. Experiments using this approach are summarized below and denoted

by the perturbation method defined above. Experiments using the standard approach, which only generate neighbors localized to the current best information vector, are prefixed with a B.

3.3 Experimental Results

To validate the solution method developed in section *3.2*, a randomly generated ROMDP is constructed. The state space is defined by two parameters *X, Y* where *X* represents the number of observations sets and *Y* represents the number of states per observation set. The total number of states is in the process is (*X***Y*). A total of *X* transition probability matrices (**P**) are generated with each element in **P** having positive probability. This results in single communicating class containing all states. Since **P** is a regular matrix, each possible transition matrix **P`** that is constructed from the state space (*S*) and action space (*A*) will again result in a regular matrix, which has a solution. Therefore every alternative in *A* is admissible for all observations sets S_k. The size of the action space is X^X. A total of 1000 problem instances are generated for evaluation.

The algorithm of Hordijk and Loeve (1994) uses successive approximations to find a local optimal policy. Since their algorithm does not have a local improvement component, the results obtained here under the policy iteration phase without perturbations serve as a benchmark for the best solution attainable by their algorithm. Similarly, the enumerative based algorithms always yield the optimal solution, but at the cost of execution time. The execution time for total enumeration is also presented to demonstrate the efficiency of the algorithm developed in this paper. The fraction of optimal solutions found for the various policy perturbation algorithms defined earlier is displayed in the graph below.

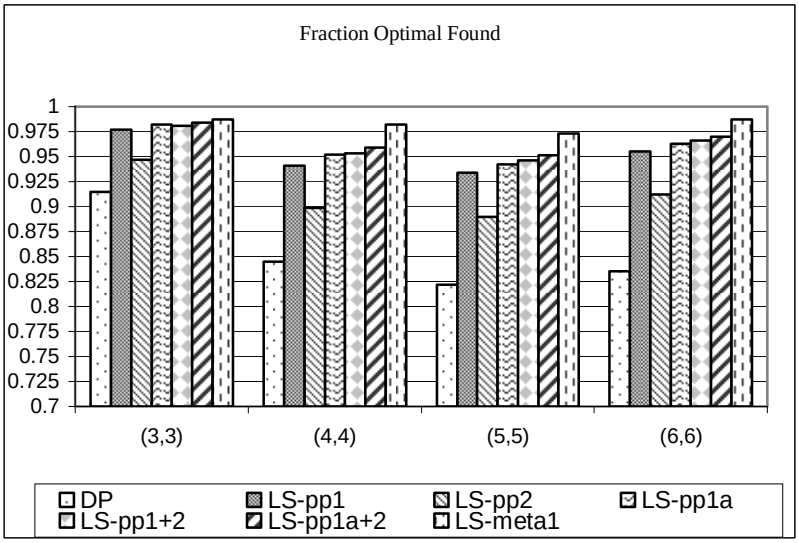


Figure 3.1 Fraction optimal solutions found over 1000 instances

In each instance, performing policy perturbations significantly improves the solution obtained from policy iteration (DP) alone. Policy iteration with perturbations (LS-ppx) significantly increase the probability of finding the optimal solution. Using LS-*pp1* alone outperforms LS-*pp2* in all cases. However, combining the two algorithms yields even better results. The results associated with LS-*meta1* outperform all other perturbation methods. This method uses multiple candidate solutions to construct neighboring polices and indicates a multi-start or genetic algorithm may be suitable for solving instances of the ROMDP.

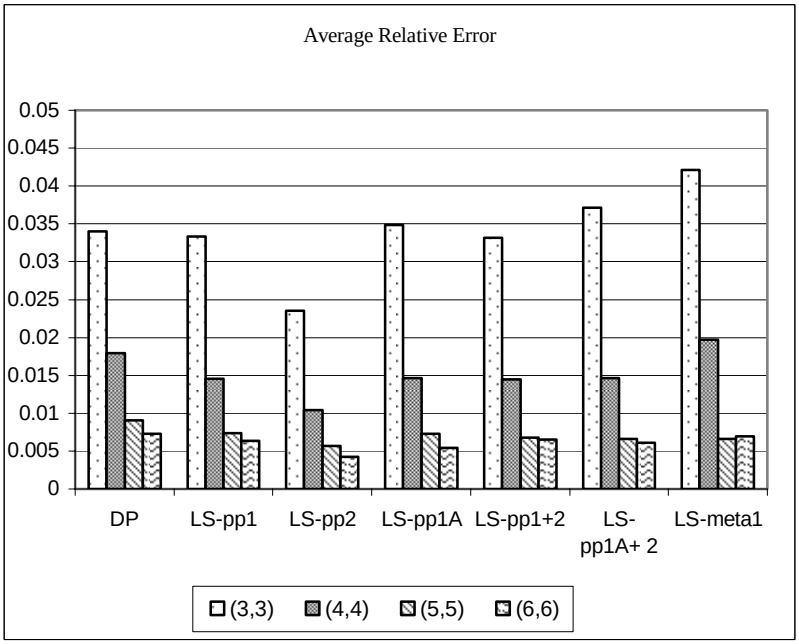


Figure 3.2 Average relative error for non-optimal solutions

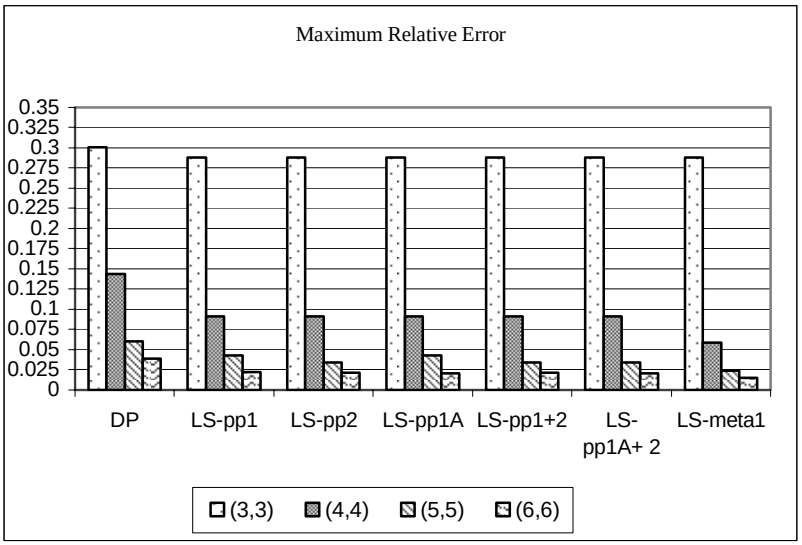


Figure 3.3 Maximum relative error for non-optimal solutions

In general, the average error for the problems where the optimal solution is not found is less than 5 percent. In addition, as the problem size increases, this error is consistently below 1 percent. The maximum relative error is displayed in Figure 3.3. This graph measures the deviation between the optimal solution and local solution found by the algorithm. For the small (3,3) problem, this deviation is as much as 30% under dynamic programming phase. As the problem sizes increase, the maximum error is below 5 percent. The execution time of each algorithm is shown in the table below. The algorithm is developed in C++ and executed on a 2.4 GHz processer with 1GB of RAM. The table displays the average execution time (in seconds) over the 1000 problems instances solved.

Table 3.1 Average execution time in CPU seconds

SIZE	DP	LS-pp1	LS-pp2	LS-pp1a	LS-pp1and2	LS-pp1A and 2	LS-meta1	Enumeration
(3,3)	7.0E-05	1.0E-04	8.0E-05	1.5E-04	2.0E-04	2.1E-04	3.7E-04	0.00135
(4,4)	0.00011	1.90E-04	0.00152	0.00032	0.00036	0.00042	0.00066	0.00369
(5,5)	0.00040	0.00065	0.00101	0.00083	0.00135	0.00152	0.00175	1.43643
(6,6)	0.00070	0.00131	0.00247	0.00177	0.00329	0.00337	0.00469	38.46965

The random problem is also solved using perturbations on the state information vector as described earlier. A total of 40 perturbations are executed before the algorithm terminates. The graphs below illustrate the results for the various problem sizes generated. For each problem

instance, the starting point for perturbation is the information vector associated with the local solution found during the policy iteration phase of the heuristic. The value of the step size ε is $1/N$, where N denotes the total number of states in the Markov process. Figures 3.4 – 3.7 display the fraction of optimal solutions found for each problem size and perturbation method.

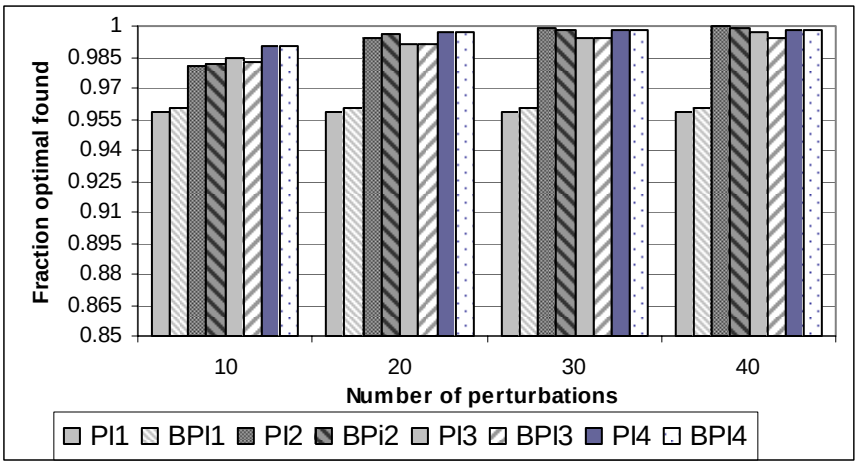


Figure 3.4 Fraction optimal found (3x3)

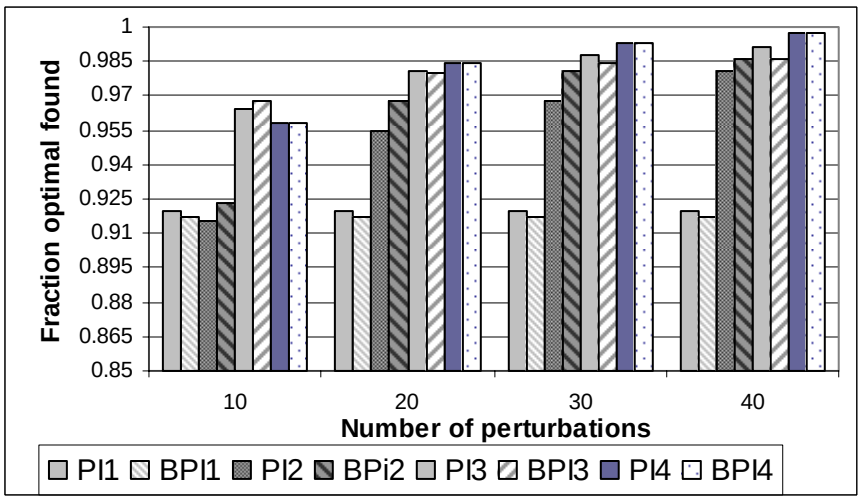


Figure 3.5 Fraction optimal found (4x4)

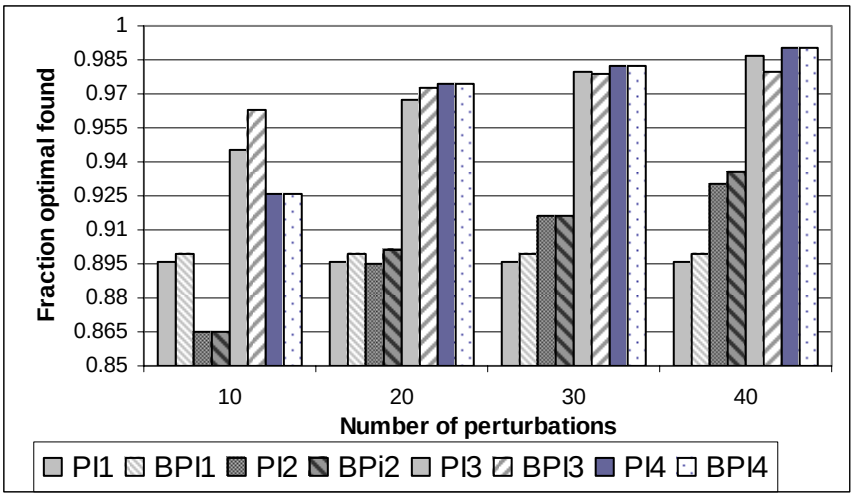


Figure 3.6 Fraction optimal found (5x5)

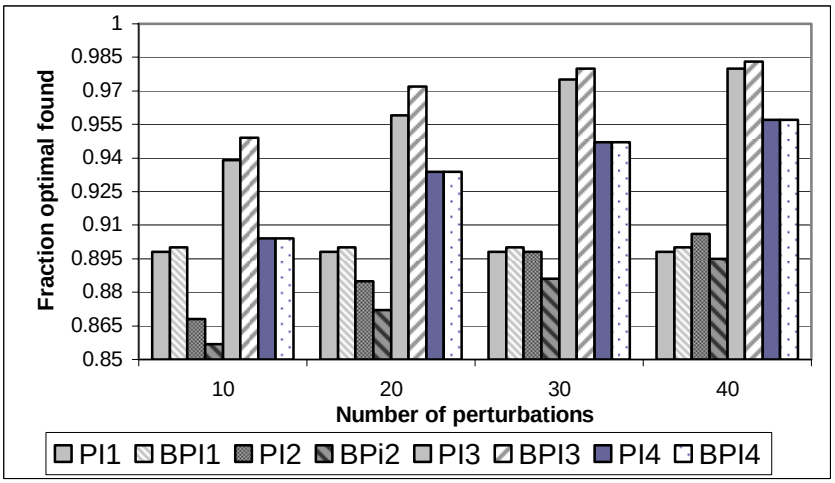


Figure 3.7 Fraction optimal found (6x6)

In all instances, PI1(BPI1) are dominated by the other perturbation methods. As the problem size increases, PI2(BPI2) is dominated as well by PI3 and PI4. When the problem size is small, PI2(BPI2), PI3(BPI3) and PI4(BPI4) are providing comparable results, with the exception of the (6x6) problem.. In that scenario, PI3 and BPI3 are the best perturbation methods. Based on these results, PI1 and PI2 can be eliminated from further consideration.

After 40 perturbations, more then 97% of the problems are being solved to optimality, which is comparable to the policy perturbation method. However, as the problem size increases, the fraction of problems solved optimally starts to decrease. This was not observed in the policy

perturbation case. When doing perturbations on the information vector, it may be necessary to make the perturbation counter a function of the problem size in order to achieve the best results. Further experimentation with the perturbation methods and termination criteria is discussed in chapter 5.

The execution time for performing 40 perturbations is shown in the table below. The execution time includes the time for performing the policy iteration phase. Perturbations using the information vector are more expensive than policy perturbations but the solutions obtained are slightly better in comparison.

Table 3.2 Average execution time in CPU seconds

SIZE	PI1	PI2	PI3	PI4
(3,3)	8.41E-03	8.24E-03	7.31E-03	7.76E-03
(4,4)	1.17E-02	0.01031	0.01046	0.01045
(5,5)	0.01959	0.01843	0.01858	0.01859
(6,6)	0.04430	0.04382	0.04402	0.04364

3.4 Conclusions

For the randomly generated problem instances, the results indicate the algorithm solves over 97% of the problems instances optimally with policy perturbations and 99% using perturbations on the information vector. Using the local improvement procedure provides significant improvement over the policy iteration phase alone. The algorithm is also very efficient as indicated by the table of execution times. Further analysis with respect to the information sharing problem is examined in the next chapter. Results for larger state and action spaces are examined in subsequent chapters.

Chapter 4 Supply Chain Model

4.1 Problem description for Inventory Information Sharing

Consider a two-stage supply chain consisting of a single retailer and single supplier sharing inventory and demand information. The retailer implements a fixed inventory control policy, while the production control policy for the supplier is to be determined from the model. The delivery lead-time is one period and therefore, orders placed at the beginning of the period are received at the end of the period. The sequence of events during a period is as follows.

1. The retailer examines his inventory and places an order.
2. The supplier receives the order and ships the available quantity from inventory. Any demand not filled from inventory is lost.
3. The supplier makes his order decision according to the decision policy.
4. Costs are calculated.
5. The retailer's order quantity is received into inventory.
6. The supplier's production quantity is received into inventory.

The objective is to measure the value of sharing inventory information using a Markov Decision Model.

The state definition is denoted by (I_s,I_r) where I_s represents the inventory level at the supplier and I_r represents the inventory level at the retailer, both observed at the beginning of the period. Full information sharing represents a completely observable process, which is modeled and solved using Howard's (1960) policy iteration algorithm. No information sharing represents a partially observable process, which is modeled and solved using the methods outlined in chapter 3. The state space is partitioned into state sets representing the observable part of the process; the supplier's inventory level. Formally, each observation set S_i contains all states (i,j) where the supplier's current inventory level is i.

Two inventory control policies for the retailer are evaluated; an order-up-to policy and an (s,S) policy. With an order–up-to policy, the retailer's order quantity is simply the difference between his inventory capacity and current on hand inventory. With an (s,S)

policy, the retailer places an order for the amount required to bring his inventory level back up to *S* when his available inventory falls below the safety stock, *s*. Based on the assumptions of the model, the transition functions for the supplier (ε_s) and retailer (ε_r) can be represented by the equations below.

$$\varepsilon_s(i,j,d) = (i - z_r)^+ + z_s$$

$$\varepsilon_r(i,j,d) = (j - d)^+ + z_r$$

Here, z_r represents the retailer's order quantity; z_s represents the supplier's production quantity; and d represents the retailer demand observed during the period. The supplier's production quantity represents the decision to be optimized in the model and can take on values $x \in [0, C_s]$ with C_s denoting the capacity at the supplier. Not all states allow the maximum production capacity, C_s, to be produced. The set of admissible production decisions is limited by the capacity available during the period. The new state transitioned to by the supplier (ε_s) after filling demand k_r and producing quantity k_s must not exceed his capacity C_s. For example, assume the maximum capacity at the supplier and retailer is 4 units and, the retailer operates under an order-up-to policy. If the current state of the system is (4,2) the set of admissible production alternatives under full information sharing is {0,1,2}, while under no information sharing, this set is simply {0}. Under a policy of no information sharing, the set of admissible actions for a given observation set is the union of all admissible actions for each state in the observation set. Given the assumptions defined for this problem, when no information is being shared the set of admissible actions, $A(k)$, for observation set S_k is $\{0,\ldots C_s\text{-}k\}$.

The per period expected supply chain cost (G_{sc}) given the current state of the system (i,j) consists of the retailer and supplier inventory holding costs with unit costs h_r and h_s respectively, as well as their penalty costs (p_r and p_s) incurred when there is insufficient inventory to meet demand.

$$G_{sc} = h_s(i - k_r)^+ + p_s(k_r - i)^+ + [(h_s + h_r)(j - d)^+ + p_r(d - j)^+]p_D(d)$$

The function $(j-d)^+$ is defined as $\max(0, j-d)$. The optimal policy minimizes the undiscounted expected per period costs over an infinite horizon.

4.2 Design of Experiment

4.2.1 Information sharing Models

Retailer demand is generated using distributions that are either randomized discrete distributions (RDD), binomial or discrete uniform. The randomized discrete distribution is created by generating random integers for a vector of length C_s+1. Each element is then divided by the vector sum to yield a number between 0 and 1. The resulting vector is a probability mass function for a distribution that takes on values between 0 and C_s. The total number of states in the Markov Decision model is $(C_s+1)*(C_r+1)$. C_s and C_r denote the capacity at the supplier and retailer respectively. For the no information sharing model, the total number of observation sets is (C_s+1), the number of states per observation set is (C_r+1) and the size of the action space is (C_s+1)!. The penalty cost for the supplier is 3 units; penalty cost for the retailer is 14 units; holding costs for both parties are 1 unit. The algorithm outlined in Chapter 3 is used to determine the optimal policy and gain associated with no information sharing and is validated via total enumeration. The optimal policy and gain under full information sharing is determined using Howard's (1960) policy iteration procedure.

The results displayed in tables 4.1 and 4.2 indicate the optimal policy (δ_s) and gain associated with no information sharing and information sharing on a small subset of problems.

Table 4.1. Policies for Inventory Sharing/Perfect Supplier/Lost Sales

C_s	C_R	DEMAND DISTRIBUTION	δ_R	δ_S NO IS	# LOCAL MINIMUMS	# OPTIMAL SOLUTIONS	# FEASIBLE SOLUTIONS (ACTION SPACE)
2	3	RDD[0,3]	Order-up-to	(2,1,0)	1	1	3! = 9
3	3	RDD [0,3]	Order-up-to	(3,2,1,0)	1	1	4! = 24
4	3	RDD [0,3]	Order-up-to	(3,2,1,0,0)	17	1	5! = 120
5	3	RDD [0,3]	Order-up-to	(3,2,1,0,0,0)	215	2	720
4	5	Uniform [0,5]	Order-up-to	(4,3,,2,1,0)	2	1	120
4	5	Uniform [0,5]	(*S*,*s*) *s*=*Cr*/2	(4,3,0,0,0)	24	24	120
7	7	Binomial	Order-up-to	(5,5,5,4,3,2,1,0)	2213	1	40,320
7	7	Binomial	(*S*,*s*) *s*=Cr/2	(6,5,4,3,0,0,0)	4438	360	40,320

Table 4.2. Inventory Sharing/Perfect Supplier/Lost Sales – Gain

C_S	C_R	GAIN OPT NIS	GAIN ALG NIS	GAIN IS	EST. % SAVINGS	ACT. % SAVINGS	%ERROR
2	3	8.47672	8.47672	5.28535	60.38%	60.38%	0
3	3	5.08768	5.08768	5.08768	0	0	0
4	3	5.08768	5.08768	5.08768	0	0	0
5	3	5.08768	5.08768	5.08768	0	0	0
4	5	7.50722	7.50722	5.61047	33.81%	33.81%	0
4	5	7.99804	7.99804	7.99804	0	0	0
7	7	1.35609	1.35609	1.33672	1.45%	1.45%	0
7	7	2.51928	2.51928	2.2939	12.84%		0

Initial results for this small subset of problems indicate the following.

1. The algorithm is finding the optimal solution in all cases where an optimal solution can be determined via total enumeration.
2. In some cases, cost savings can be achieved with inventory information sharing. The specific control rules by which this savings is achievable is discussed in subsequent chapters for larger models.
3. In this limited experiment, the demand distribution, retailer policy, and supplier capacity affects the value of information sharing. Table 4.3 illustrates the impact capacity has on the value of information sharing. Since the optimal solution for no information sharing could not be determined via total enumeration, the estimated savings serves as an upper bound on the value of information sharing. The actual value could be less.
4. The results indicate the value of information sharing is always positive. It can never be more beneficial to not share information. Lemma 4.1 further explains why this result is true.

Table 4.3. Capacity influence on Inventory Sharing/Perfect Supplier/Lost Sales instances

C_S	C_R	DEMAND DISTRIBUTION	δ_R	δ_S NO IS	δ_S WITH IS	δ_S OPT NO IS	EST. SAVINGS	# OF ALT. POLICIES
5	15	Discrete Uniform [0,15]	*(S,s)s=5*	16.7417	16.7417	16.7417	0	6!
7	15	Discrete Uniform [0,15]	*(S,s)s=5*	15.3037	15.3037	15.3037	0	8!
9	15	Discrete Uniform [0,15]	*(S,s)s=5*	13.8656	13.8656		0	10!
11	15	Discrete Uniform [0,15]	*(S,s)s=5*	12.4276	12.4276		0	12!
13	15	Discrete Uniform [0,15]	*(S,s)s=5*	11.6047	11.5082		0.832%	14!
15	15	Discrete Uniform [0,15]	*(S,s)s=5*	11.3890	11.0318		3.136%	16!
18	15	Discrete Uniform [0,15]	*(S,s)s=5*	11.3890	11.0318		3.136%	19!
20	15	Discrete Uniform [0,15]	*(S,s)s=5*	11.3890	11.0318		3.136%	21!
23	15	Discrete Uniform [0,15]	*(S,s)s=5*	11.3890	11.0318		3.136%	24!
25	15	Discrete Uniform [0,15]	*(S,s)s=5*	11.3890	11.0318		3.136%	26!

Lemma 4.1: If g^o is the optimal gain of the completely observable process and g^r is the optimal gain of the restricted observation process then $g^o \le g^r \left(g^o \ge g^r\right)$ for a cost minimizing (maximization) Markov decision model. This implies the cost reduction achievable with information sharing will always be greater than or equal to zero $\left(g^r - g^o = \Delta_{is} \ge 0\right)$. It will never be more beneficial to not share information. It will either be beneficial $\left(\Delta_{is} > 0\right)$ or not $\left(\Delta_{is} = 0\right)$.

Proof:

Let Π^D represent the set of all deterministic policies and Π^R represent the set of all admissible policies for the no information sharing model. ($\prod^R \subseteq \prod^D$). Suppose policy α^o with gain g^o is found to be optimal for the completely observable MDP and policy α^r with gain g^r is found to be optimal for the ROMDP. Also, assume $g^r < g^o$. Let superscripts *r* and *o* indicate quantities relevant to policies α^r and α^o. If policy α^r is better than α^o, then from Howard's (1960) proof of convergence, it must be true that

$$q_i^r + \sum_j p_{ij}^r v_j^o \le q_i^o + \sum_j p_{ij}^o v_j^o \quad \forall i = 1...N$$

and

$$q_i^r + \sum_j p_{ij}^r v_j^o < q_i^o + \sum_j p_{ij}^o v_j^o \quad \text{for at least one } i.$$

This implies that policy α^o could not be optimal for the completely observable MDP and policy α^r would be found during the policy improvement step of Howard's (1960) policy iteration algorithm. Therefore, policy α^r with gain g^r is either optimal for the completely observable MDP and thus $\Delta_{is} = 0$, or there exists a some policy *b* with gain $g^b < g^r$ which is optimal for the completely observable MDP and $\Delta_{is} > 0$. Q.E.D.

It is easy to see the optimal policy under no information sharing is a subset of the admissible policies for the completely observable MDP.

4.2.2 Randomly Generated Models

4.2.2.1 Solution by policy perturbation

In order to test the performance of the algorithm on a large number of information sharing problems, 1000 instances of the supply chain problem are generated. The first case uses the same cost structure defined in section 4.1.1 using random demand distribution with fixed order-up to policy employed by the retailer. The results associated with this experiment are summarized in figures 4.1 – 4.3.

A second set of experiments is executed using binomial demand distribution with an (*s*,*S*) policy employed by the retailer with parameters C_s and $C_s/2$. The results are summarized in Figures 4.4 – 4.6. The fraction of optimal solutions found and the relative error measure performance of the algorithm with respect to total enumeration. The relative error is defined as $\left(\tilde{g} - g^*\right)/g^*$, where $\tilde{g}$ is the value obtained by the heuristic and g^* is the optimal value obtained via total enumeration.

In comparing the two sets of experiments, there is a large disparity in the fraction of optimal solutions found. When a problem instance is generated using randomly generated customer demand distribution and order-up-to retailer policy, over 98% of the problem instances are solved optimally irrespective of the problem size. When the binomial demand distribution is used with (*s*,*S*) retailer policy, only 98% of the problems are solved optimally when the problem size is small. The fraction of optimal solutions

founds declines rapidly as the problem size increases. Similar trends exist for the average relative error and maximum relative error. Figure 4.6 illustrates that the maximum relative error increases as the problem size increases, while figure 4.3 shows maximum relative error decreasing. It is evident from this experiment that the choice of parameters (demand distribution, retailer policy) vastly affects the performance of the problem. Chapter 5 discusses this discrepancy in detail and describes an improvement made to the algorithm to mitigate this problem.

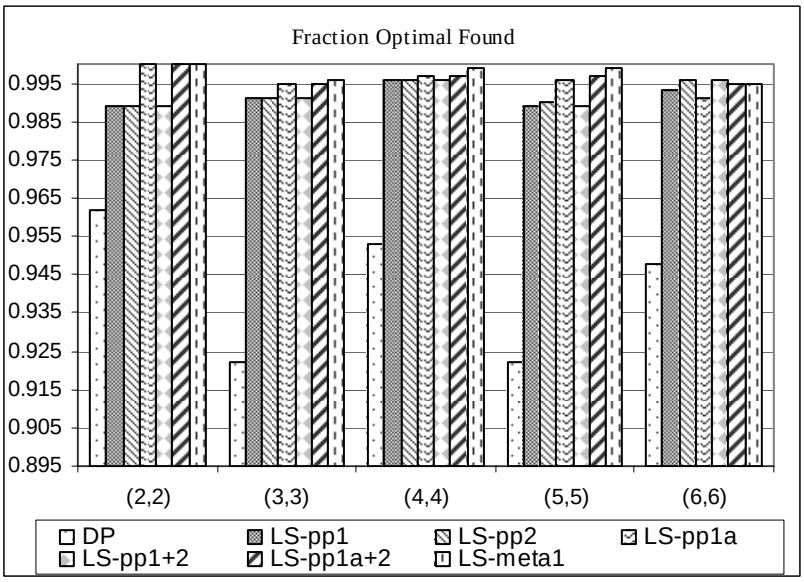


Figure 4. 1 Fraction Optimal found – Randomized discrete distribution

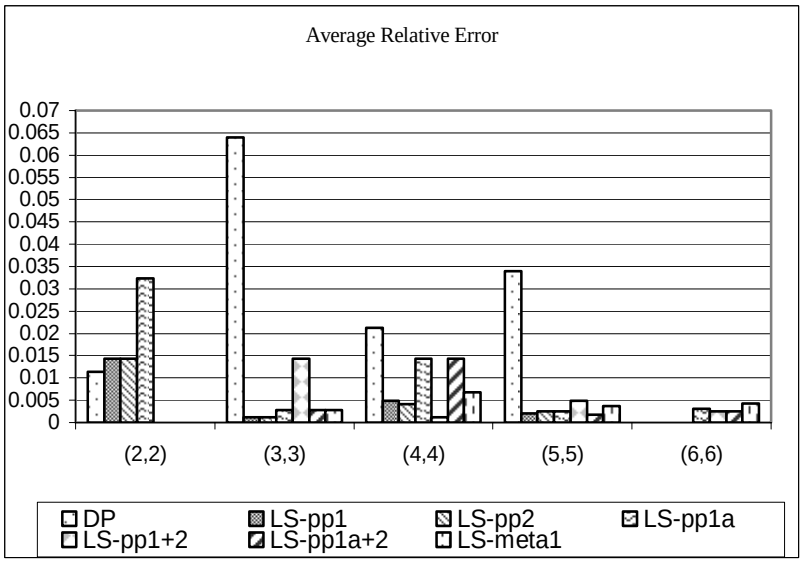


Figure 4. 2 Average Relative Error – Randomized discrete distribution

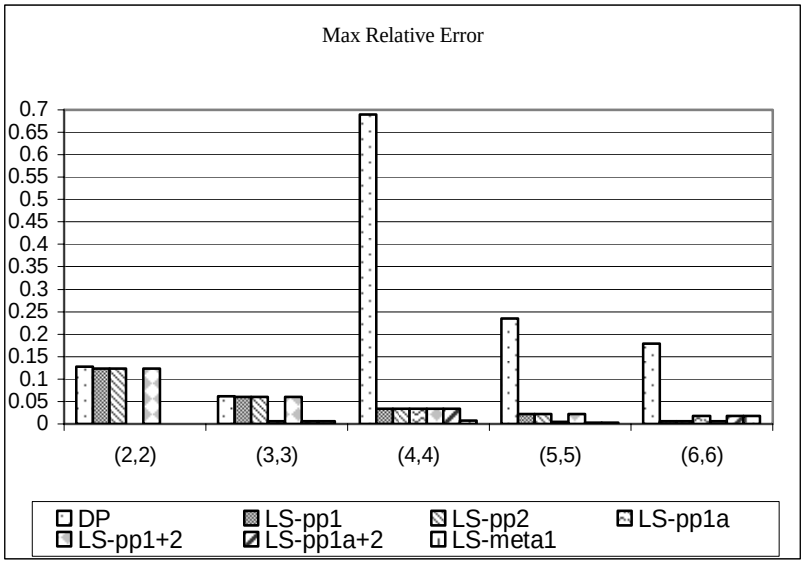


Figure 4. 3 Maximum Relative Error – Randomized discrete distribution

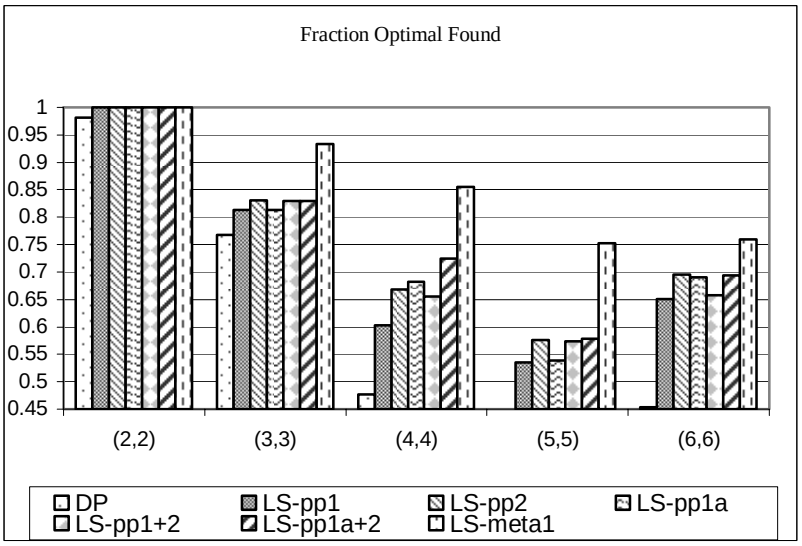


Figure 4. 4 Fraction optimal found - Binomial demand distribution

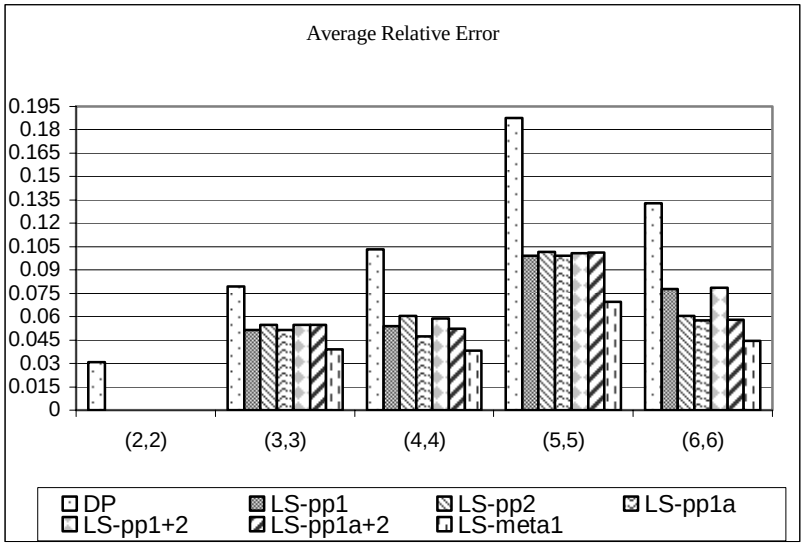


Figure 4. 5 Average Relative Error - Binomial demand distribution

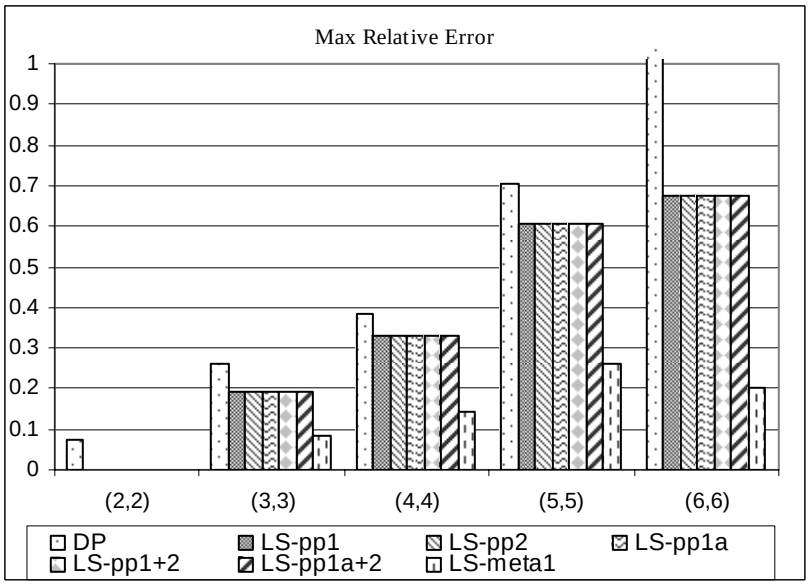


Figure 4. 6 Maximum Relative Error - Binomial demand distribution

4.2.2.2 Solution from Perturbations on information vector

The information sharing models defined in section 4.2.2.1 are also solved using perturbations on the information state vector as described in chapter 3. The results for both random demand and binomial demand are summarized in figures 4.7 – 4.15. As observed in the prior section, the fraction of optimal solutions found is significantly less

when using binomial demand and (*s*,*S*) policy for the retailer. The fraction of optimal solutions found is similar to the fraction found during policy perturbations and in some cases is less. For example, in the (3,3) binomial demand case, policy perturbation solves 93% optimally, while perturbations on the information vector only solves 90%. Similarly, with the (5,5) binomial demand case, the best policy perturbation solves 75% optimally while the best information vector perturbation only solves 70%.

In the supply chain problem, information vector perturbation PI2 and BPI2 significantly outperform all of the other perturbation methods. For the (3,3) random demand case, after 40 perturbations PI2 solves 98.8%, while PI3 solves 94%. Similarly for the (4,4) binomial demand case, PI2 solves 84%, while PI3 and PI4 solve 64%. This result is different than what was observed for the random problems in chapter 3. Recall for those experiments, PI2 and BPI2 are dominated by PI3 and BPI3. As in the randomly generated problems of chapter 3, PI1 and BPI1 provide no significant improvement over the solution obtained from the policy iteration phase of the heuristic. Therefore, these methods are eliminated from further consideration. Based on the results obtained in the supply chain problem, it appears the structure of the problem affects the selection of the dominating perturbation method. This is examined further in chapter 5.

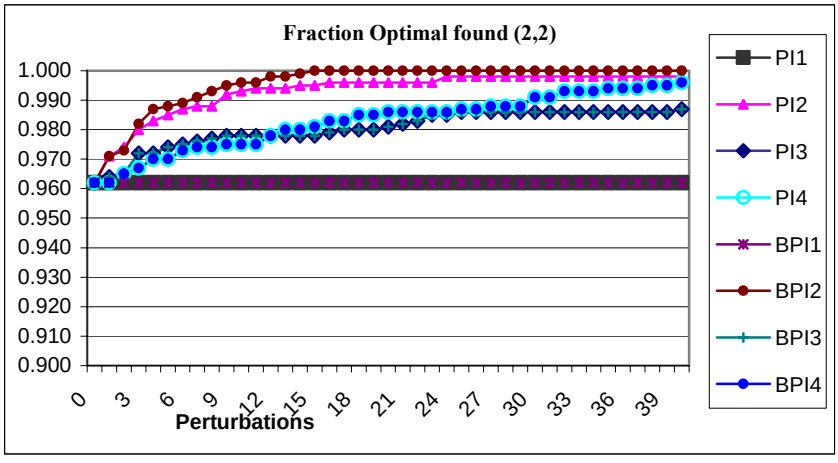


Figure 4. 7 Fraction optimal Found – Randomized discrete distribution (2,2)

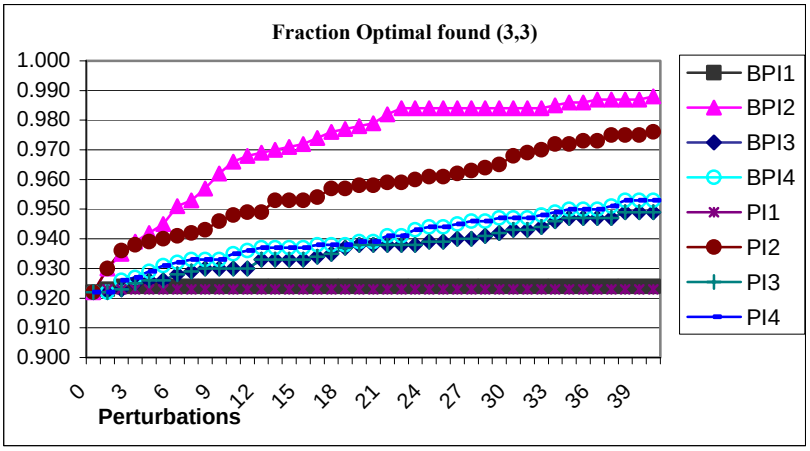


Figure 4. 8 Fraction optimal found – Randomized discrete distribution (3,3)

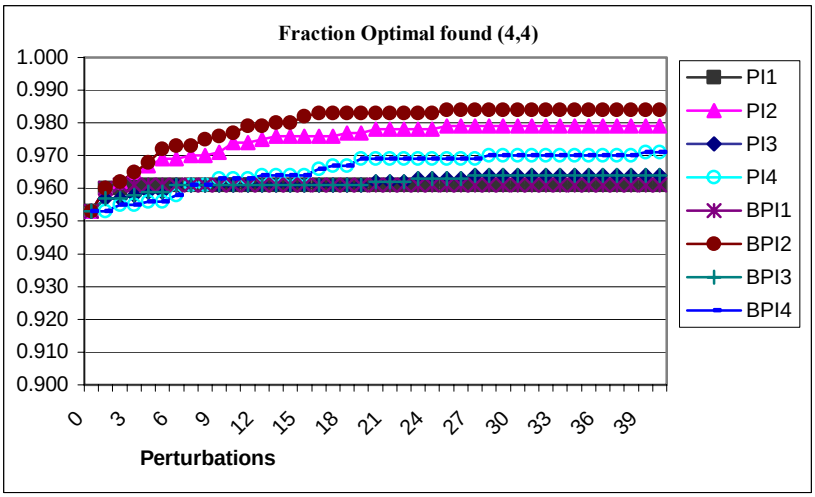


Figure 4. 9 Fraction optimal found – Randomized discrete distribution (4,4)

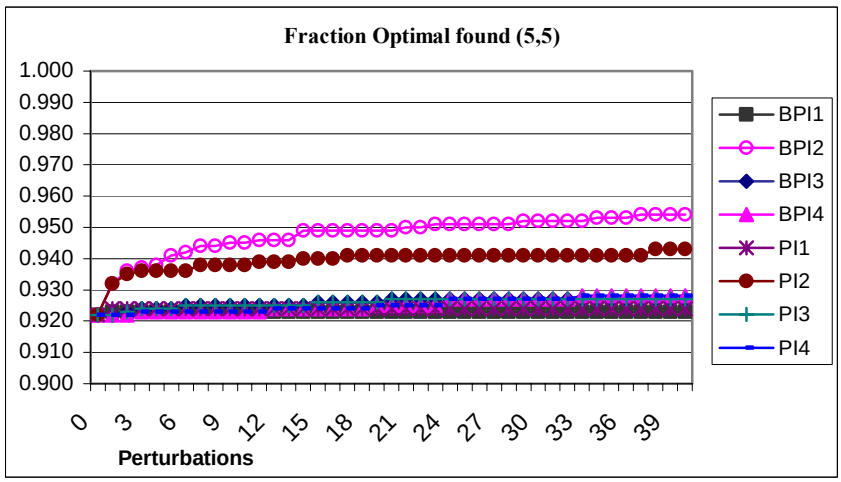


Figure 4. 10 Fraction optimal found – Randomized discrete distribution (5,5)

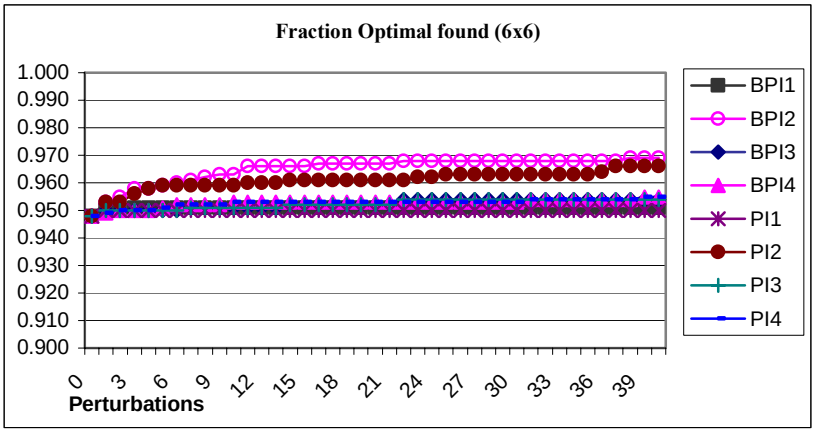


Figure 4. 11 Fraction optimal found – Randomized discrete distribution (6,6)

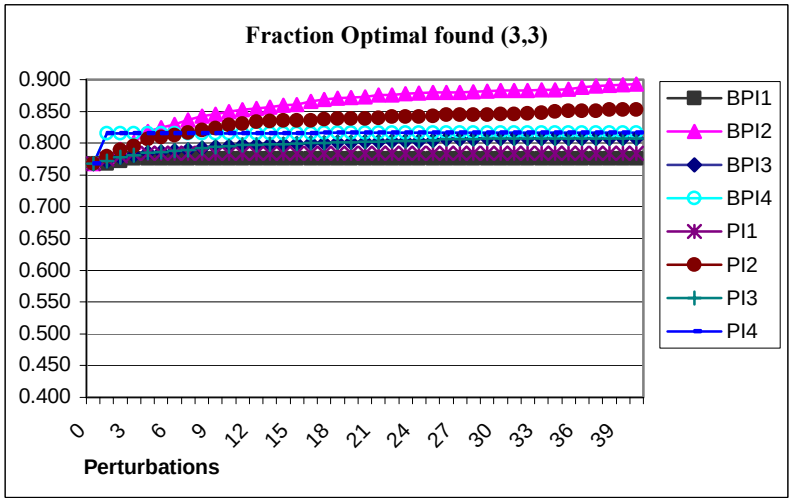


Figure 4. 12 Fraction optimal found - Binomial demand distribution (3,3)

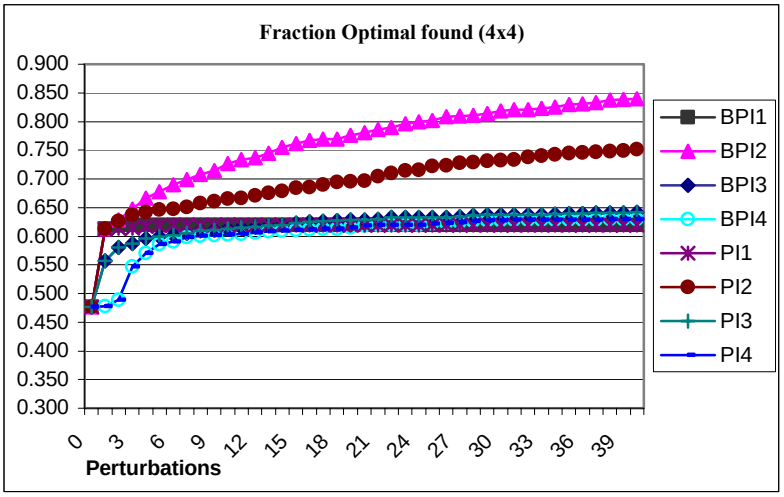


Figure 4. 13 Fraction optimal found - Binomial demand distribution (4,4)

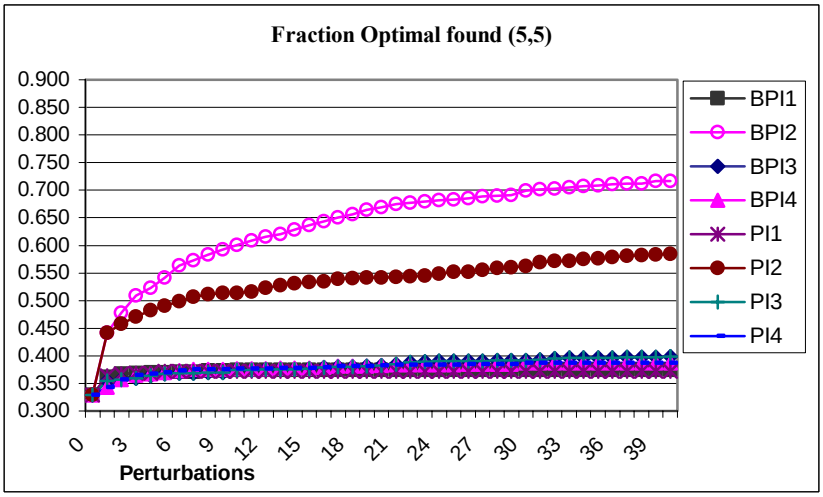


Figure 4. 14 Fraction optimal found - Binomial demand distribution (5,5)

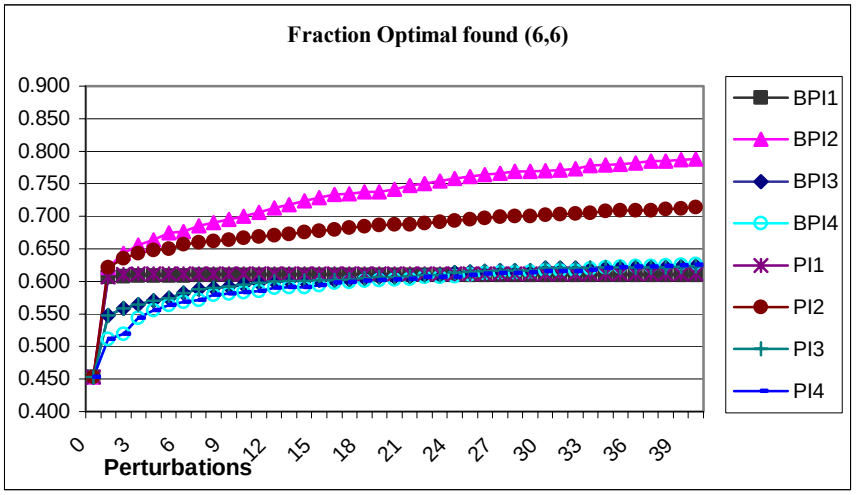


Figure 4. 15 Fraction optimal found - Binomial demand distribution (6,6)

4.3 Measuring the value of information sharing

Using the small problems generated above, we summarize the value of information sharing results to identify any trends that may hold when larger problems are solved. The table shows the number of problems out of the 1000 binomial demand (*s*,*S*) instances generated that had value in information sharing. An interesting observation is that cost savings with information sharing is only achieved when the Coefficient of Variation (C_v) of demand meets or exceeds some critical value. Table 4.4 summarizes the critical value determined during experimentation. For example, in the (2,2) problem, there are a total of 429 problems that have value in information sharing. This cost savings occurred on problems where the $C_v \geq 0.795$. As the problem size increases, the minimum value of C_v decreases and there are more problems that have value in information sharing. However, the average relative cost reduction with information sharing is getting smaller. Recall, this benefit is measured in terms of the total supply chain costs.

Although we are more interested in the policy structure for the information sharing case, there are some interesting observations for the optimal policy structure in the case of no information sharing which are based on the value of C_v. For the (2,2) problem, when $0.795 \leq C_v < 0.89$, the optimal policy for no information sharing is an order up to policy. When $0.890386 \leq C_v < 1.68983$, the optimal policy is a fixed lot size policy. When $1.68983 \leq C_v < 17.5667$, the optimal policy is a base stock policy with

critical base stock level set to C_s-1. Similar results are obtained at different problem sizes and different bounds on C_v. When the mean demand is close to or less than one (about 1.3 in 4x4 case) the optimal policy for NIS is base stock with critical value of 1. Very large values of the relative cost increase with information sharing also occur when the mean demand is less than 1. This result shows as mean demand decreases, C_v increases and there is value in info sharing, which may occur with slow moving items.

Table 4.4. Information Sharing Summary for Binomial demand (s,S) problem

				STRUCTURE OF NIS POLICY				
SIZE	# PROBLEMS WITH VALUE IN IS $(\Delta_{is} > 0)$	AVERAGE Δ_{is}	C_v MIN THRESHOLD FOR Δ_{is}	ORDER-UP TO OPTIMAL	BASE STOCK OPTIMAL	FIXED LOT SIZE OPTIMAL	(S,1)	OTHER
(2,2)	429	2.66992	0.795	55	157	217	0	0
(3,3)	565	1.81593	0.488688	48	95	76	0	346
(4,4)	683	1.01654	0.336988	58	34	217	183	0
(5,5)	757	0.953737	0.260369	57	22	0	0	678
(6,6)	805	0.70409	0.208352	63	20		121	

The results for the random demand, order-up-to policy do not possess a readily identifiable correlation between C_v and value in information sharing as in the binomial case. Further discussion on the structure of the optimal policy (with and without information sharing) and average cost savings is discussed in chapter 7, when larger problems are examined.

Chapter 5 Sensitivity Analysis

5.1 Overview

In this chapter, we analyze the performance of the ROMDP policy iteration/policy perturbation heuristic as parameters associated with the problem instance are varied. This analysis will serve as a means of identifying the perturbation strategy which yields the best results independent of the parameters of the problem being solved. Recall in the randomly generated problems of chapter 3, the heuristic solves over 99% of the problems optimally. This however, is not the case in chapter 4 where the supply chain problem is introduced. When the demand distribution is a randomly generated discrete distribution, over 98% of the problems are solved optimally irrespective of the problem size. In contrast, when the demand distribution is binomial, 100% of the problems are solved optimally for the (2,2) case and then performance decays as the solution space increases. These experiments illustrate how the performance (measured in terms of fraction of optimal solutions found) varies based on the problem parameters.

We first begin with sensitivity analysis on the policy perturbation method by varying problem size, demand distribution, supplier penalty cost, and retailer order policy. Similar analysis is performed for perturbations based on the information vector. We conclude by identifying the best strategy based on the results obtained via experimentation.

5.2 Sensitivity analysis with policy perturbation

5.2.1 Experimental Design

Consider the supply chain information sharing model described in chapter 4 where supplier capacity and retailer capacity equal 4. There are a total of 25 states in this model with a policy space of size 5!. All costs are held fixed except the penalty cost of the supplier, which is changed to roughly approximate 0, $\frac{1}{4}p_r$, $2p_r$ and $4p_r$. The quantity p_r denotes the penalty cost of the retailer. Demand distributions examined are binomial and randomly generated discrete distributions. The retailer order policy is either base stock with base stock level C_r or (s,S) with parameter s equal to $C_r/2$ and S equal to C_r.

From both the randomly generated and supply chain problems, the algorithm *Ls-meta1* provides the best results over all policy perturbation methods examined. Therefore, further analysis will consider only this perturbation strategy. Recall *Ls-meta1* uses multiple candidate solutions to construct neighboring policies.

In the previous experiments, a myopic policy is used as the initial starting point. The impact of starting from a randomly generated policy is also examined as well as the impact of using random restarts.

5.2.2 Penalty Cost Analysis

Figures 5.1 – 5.3 indicate the fraction of optimal solutions found as a function of the supplier penalty cost. In each problem set, the external demand and retailer policy are held constant. Figure 5.1 represents the results when the retailer demand distribution is a randomly generated discrete distribution and a base stock policy is used. As the penalty cost increases, the fraction of optimal solutions found increases. The best performance is achieved when the penalty cost is the highest (p_s=50). In this case, policy iteration alone solves all of the problem instances optimally without the use of perturbations. Figure 5.2 illustrates results when an (*s*,*S*) policy is used in place of a base stock policy. In this case, over 95% of the problems are solved optimally when the penalty cost is 14 or higher. The worst results are achieved in both policy iteration and policy perturbation when the demand is binomially distributed and an (*s*,*S*) policy is used by the retailer (Figure 5.3). The policy perturbation greatly increases the solution obtained from policy iteration (88% increase when p_s=3). However, the perturbation strategy is not sufficient to yield performance comparable to that of figure 5.1.

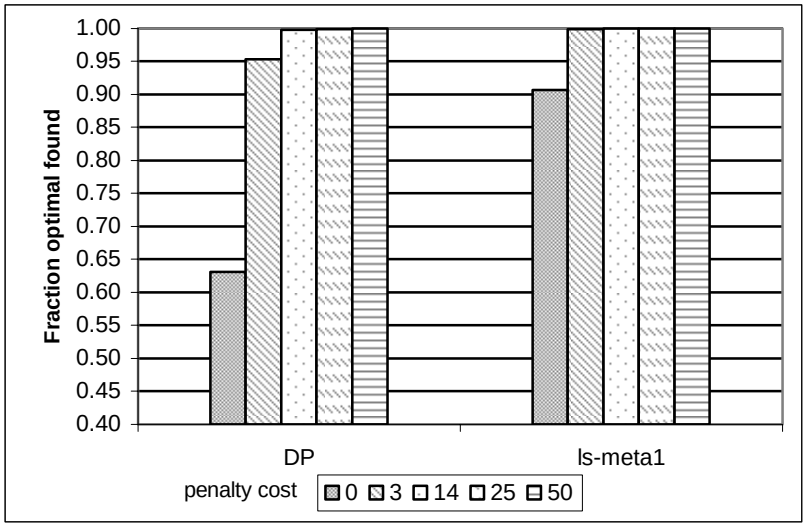


Figure 5. 1 Randomly Generated Discrete Distribution, Base Stock Policy for retailer

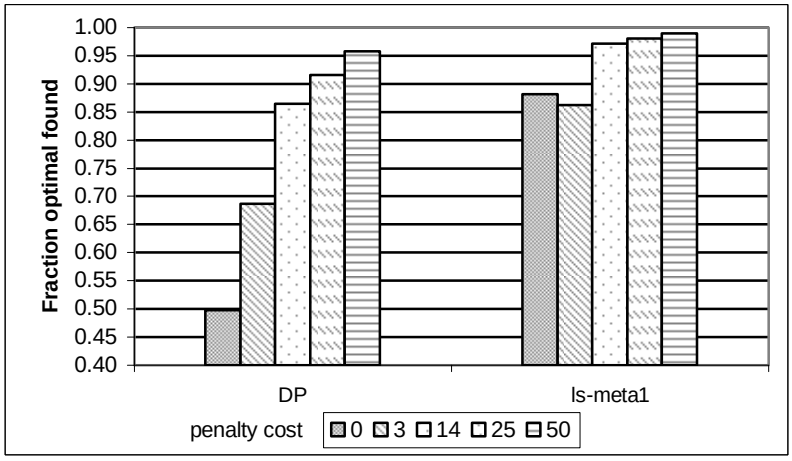


Figure 5. 2 Randomly Generated Discrete Distribution (*s*,*S*) Policy for Retailer

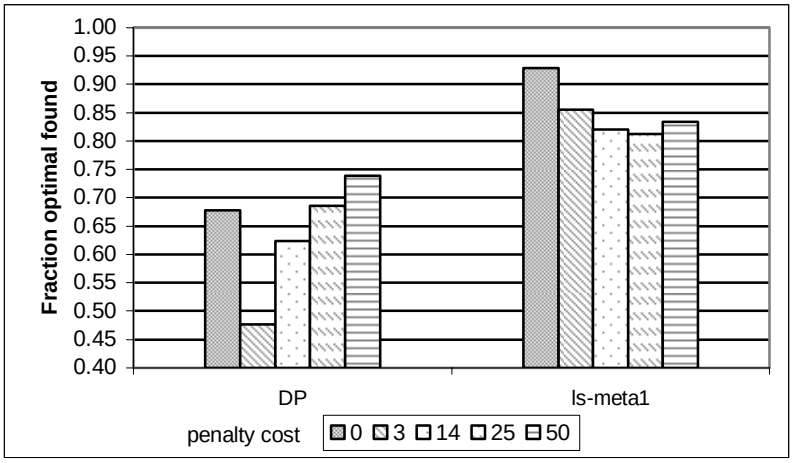


Figure 5. 3 Binomial Demand, (*s*,*S*) Policy for Retailer

Since the surface of the objective function is too complex to graph, we examine the distribution of the local minima to gain insight into the underlying surface of the problem being solved. Using the definitions defined in chapter 3, all local minima of a given problem instance are determined during total enumeration. Figures 5.4 – 5.6 display the histogram of the local minima associated with the demand and policy characteristics of figures 5.1 and 5.3.

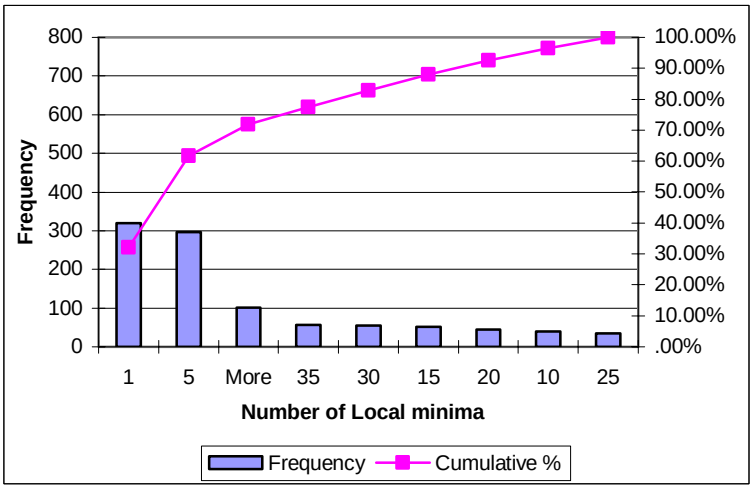


Figure 5. 4 Randomly Generated Discrete Distribution, Base Stock Policy, Penalty Cost of 0

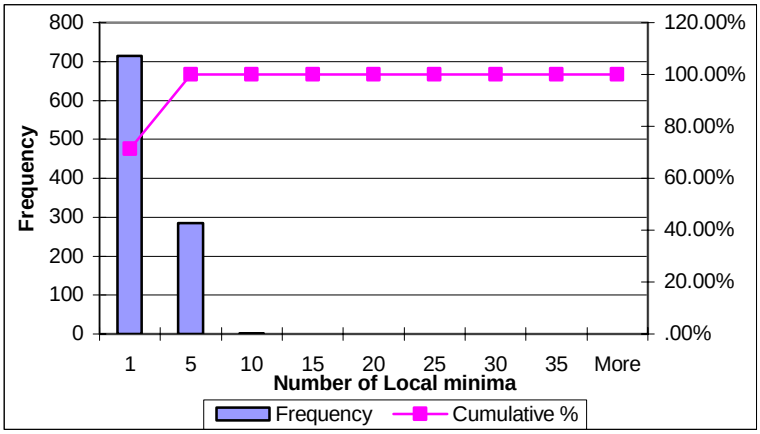


Figure 5. 5 Randomly Generated Discrete Distribution, Base Stock Policy, Penalty Cost of 50

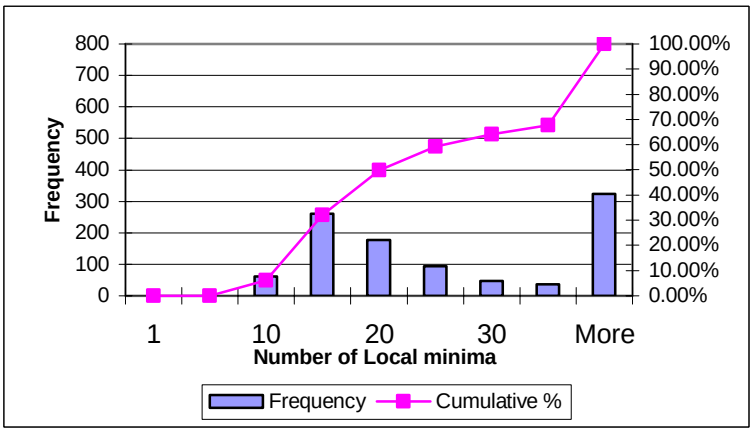


Figure 5. 6 Binomial Demand, (*s*,*S*) Policy, Penalty Cost of 3

The graphs summarize the total number of local minima per instance out of the 1000 problem instances generated. Figure 5.5 shows when the demand distribution is randomly generated, a base stock policy is used by the retailer, and the penalty cost for the supplier is 50, over 70% of the problem instances generated will have 1 local minimum. Therefore, the local minimum is the global minimum and will be found during the policy iteration phase without the use of perturbations. The rest of the problem instances have between 2 and 5 local minima.

In contrast, when the demand is binomially distributed and an (*s*,*S*) policy is used by the retailer, the number of problems with 1 local minimum is significantly smaller. Since there are several local minima, the underlying surface of the solution space may be 'hilly' and the optimal solution difficult for the heuristic find. Each local minimum can

be a potential stopping point for the policy iteration phase. Figures 5.7 and 5.8 provide a closer comparison of the effect of the retailer policy on the solution space when the retailer demand is binomially distributed.

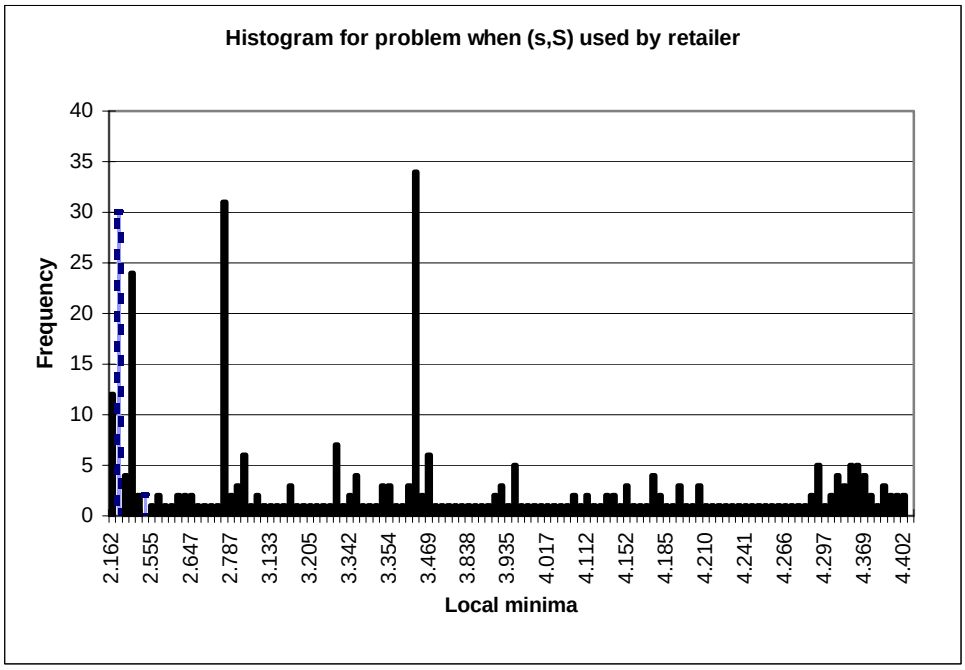


Figure 5. 7 Binomial Demand, (*s*,*S*) Policy, Penalty Cost of 3

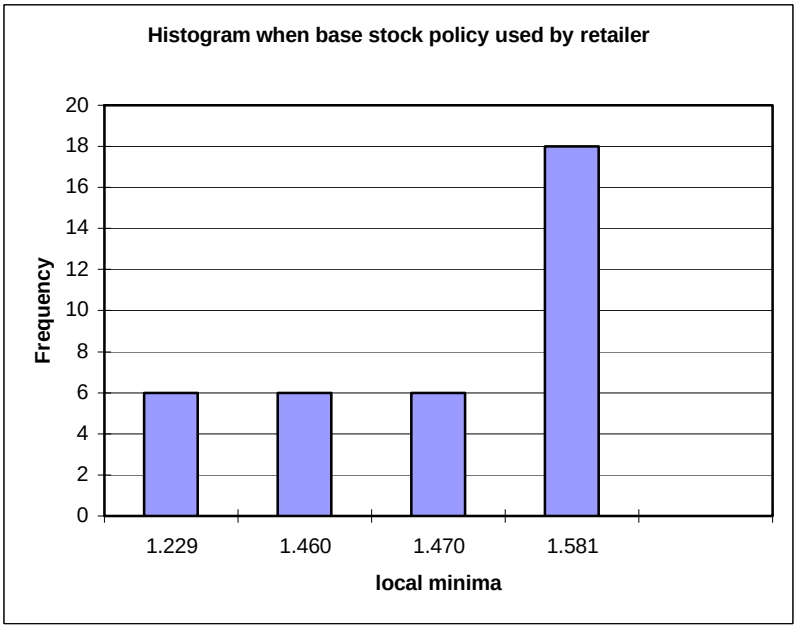


Figure 5. 8 Binomial Demand, (*s*,*S*) Policy, Penalty Cost of 3

The histograms of the local minima indicate that changing the demand and retailer order policy, greatly influence the solution space and number of local minima that exist, thus affecting the ability of the heuristic to find the unique optimal solution.

5.2.3 Retailer policy analysis

Figures 5.9 and 5.10 illustrate the variation in performance (keeping penalty cost constant) with respect to the demand distribution and the retailer order policy. In the figures below, *out* denotes order up to policy and *ss* denotes (*s*,*S*) policy.

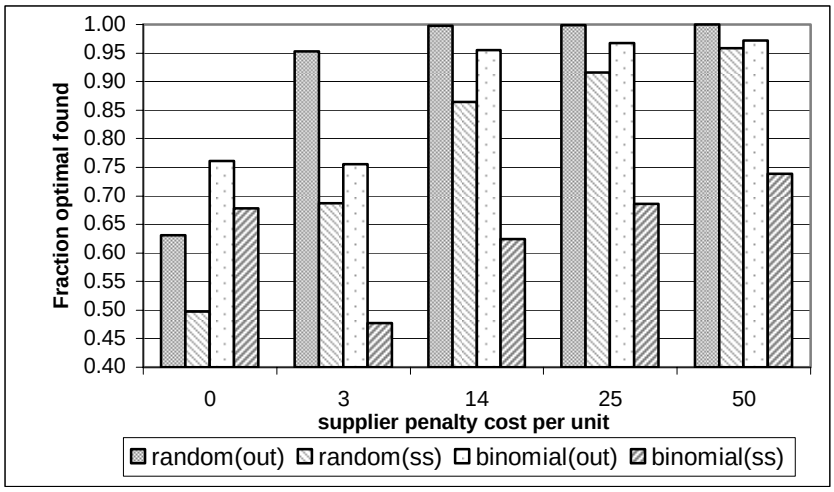


Figure 5. 9 Policy Iteration Performance (No Perturbation)

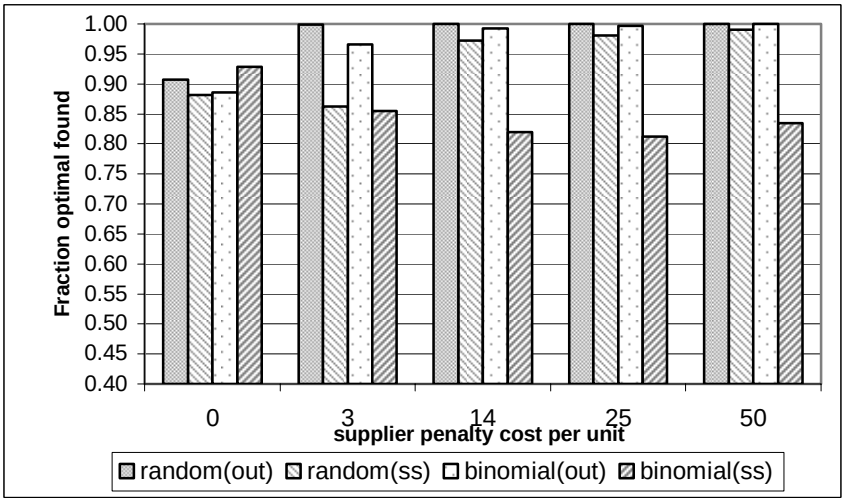


Figure 5.10 Policy Iteration - Perturbation Performance

Again, each parameter variation yields different results. When an order up to policy is used by the retailer, over 90% of the problems are solved optimally, getting up to 100% in some cases for policy iteration alone. Again, this can be attributed to the structure of the solution space for the different problems under study. The combination of binomial demand and (*s*,*S*) policy results in the most changes to the solution space and thus performance of the heuristic. The local improvement procedure (policy perturbation) definitely improves the solution obtained from policy iteration, but it also falls below 90% for cases of binomial demand and (*s*,*S*) policy.

5.2.4 Effect of initial policy

In all of the problems examined thus far, the algorithm is started with a myopic policy. Starting with a random policy is also examined and in some instances yields better results. This is exhibited in the case with random demand. In the following graphs, *r* denotes random policy start and *m* denotes myopic policy start. When p_s equals 0 or 3, the myopic policy starting point yields better results for the policy iteration-perturbation heuristic. The other scenarios achieve better results when a randomly generated starting policy is used. However, the clear best starting point is not certain when using binomial demand. In some cases, the myopic policy does lead to the optimal solution, but overall the graph shows that starting with the random policy yields slightly better results. Therefore, we examine the effect of random restarts, letting the first policy be the myopic policy with subsequent policies generated randomly.

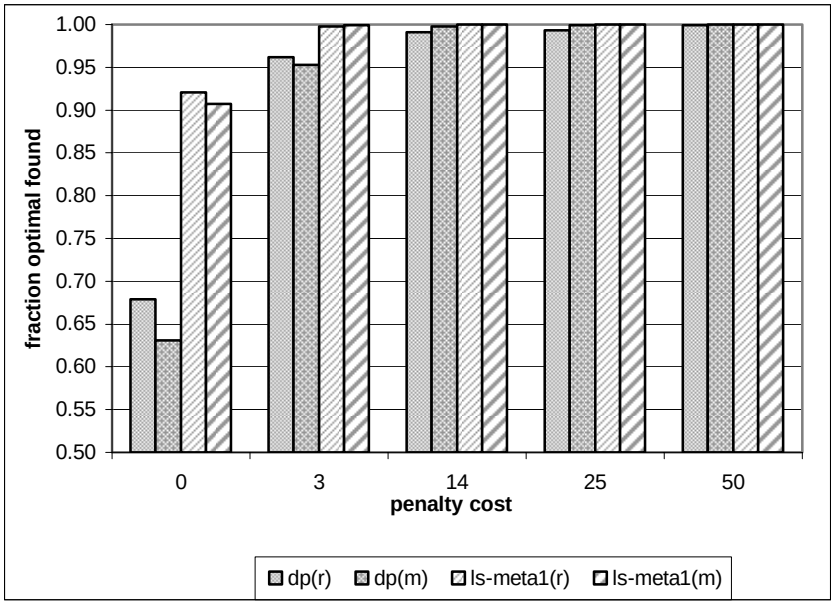


Figure 5. 11 Randomized Discrete Distribution, Base Stock policy for retailer

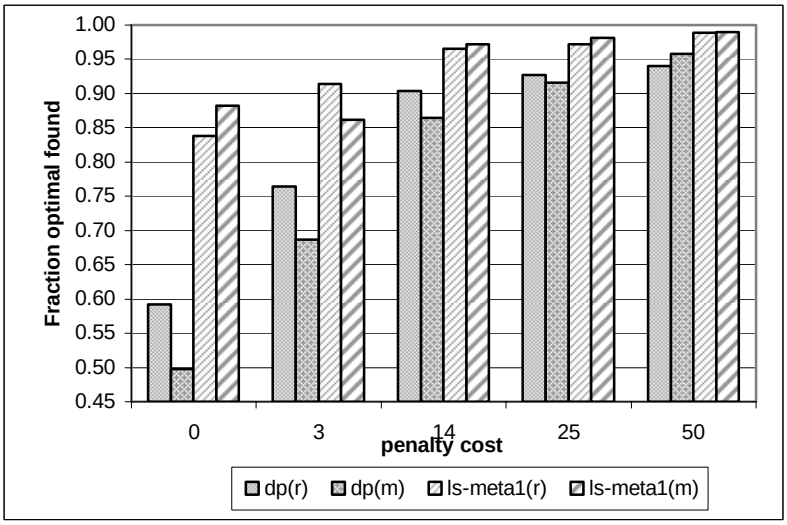


Figure 5. 12 Randomized Discrete Distribution, (*S*,*s*) Policy for Retailer

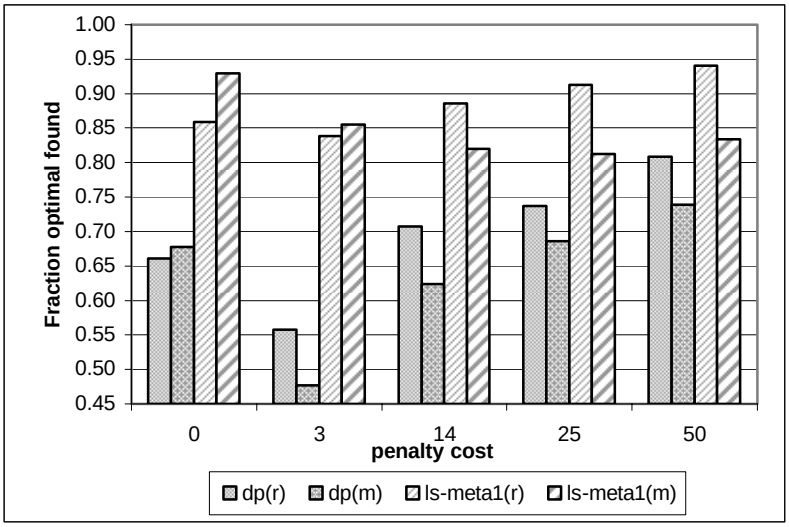


Figure 5. 13 Binomial Demand Distribution, (*s*,*S*) Policy for Retailer

5.2.2.5 Random restarts

In the previous sections, it is shown that different problem characteristics affect the ability of the heuristic to find the optimal solution. The goal is to find some strategy that consistently provides good results independent of the problem characteristics. Therefore, we study the effect of random restarts on the performance of the algorithm. We consider the extreme cases of performance; binomial distribution with (*s*,*S*) retailer policy, binomial distribution with order-up-to retailer policy, and a randomly generated discrete distribution with (*s*,*S*) policy. These problem parameters are summarized in table 5.1. Recall without restarts, when the retailer follows an (*s*,*S*) policy with binomially distributed demand and supplier penalty cost of 3, the heuristic solves 85.5% of the problems optimally. For the randomized discrete distribution with (*s*,*S*) policy and supplier penalty cost of 0, the heuristic solves 88.2% optimally. For a binomial distribution with an order up to policy and supplier penalty cost of 25, the heuristic solves 99.7% optimally. Figures 5.14, 5.16 and 5.18 display the fraction of optimal solutions found when random restarts are used. In the random restart strategy, the starting policy is always the myopic policy and each subsequent policy is randomly generated.

Table 5. 1 Supply Chain Problem Parameters

Problem number	Demand	Retailer policy	p_s
P1	Binomial	(*S*,*s*)	3
P2	Binomial	Order up to	25
P3	Random	(*S*,*s*)	0

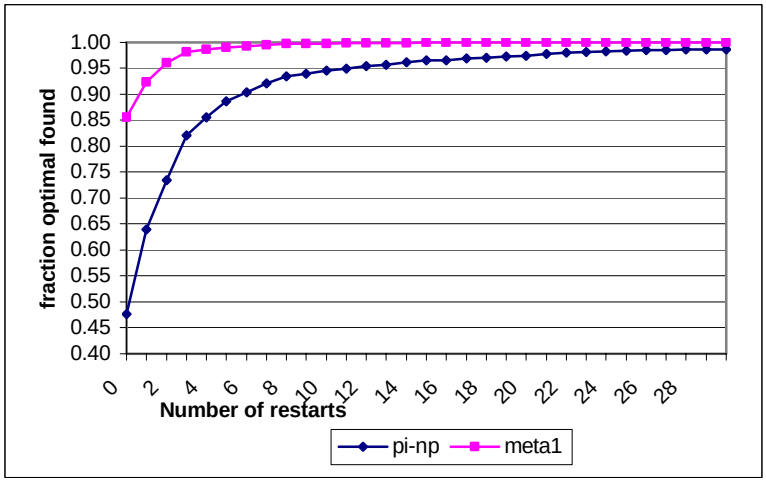


Figure 5. 7 Fraction Optimal Found for Problem P1

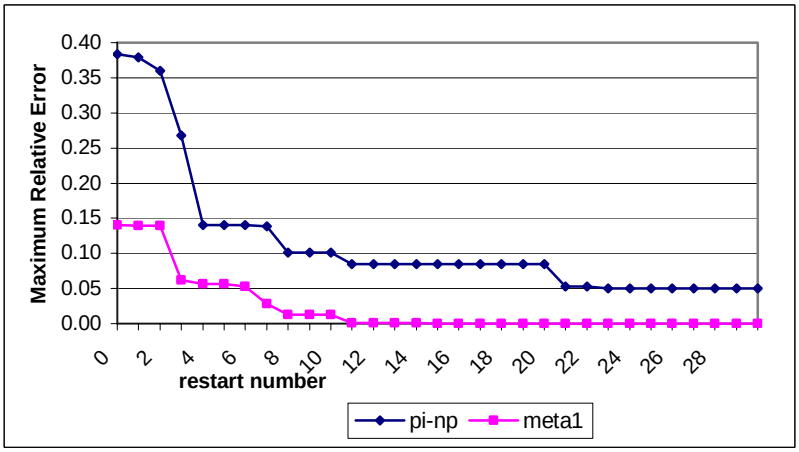


Figure 5. 8 Maximum Relative Error for Problem P1

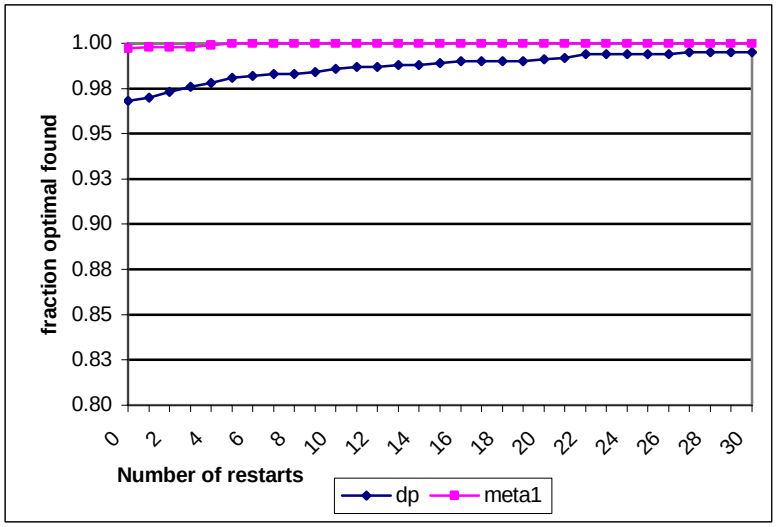


Figure 5. 16 Fraction Optimal Found for Problem P2

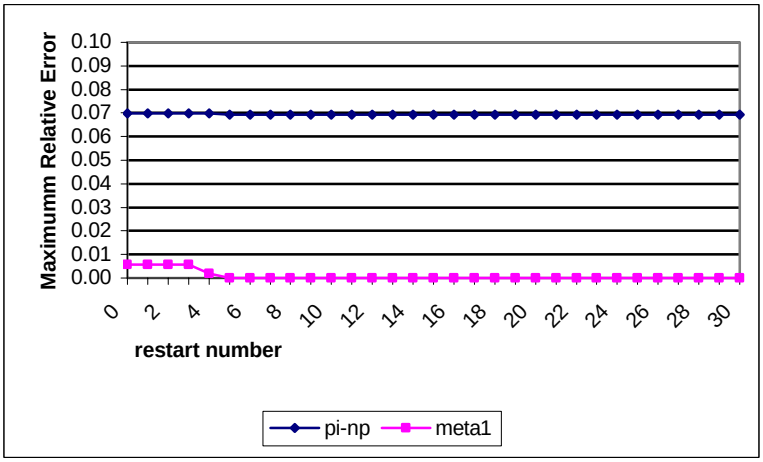


Figure 5. 9 Maximum Relative Error for Problem P2

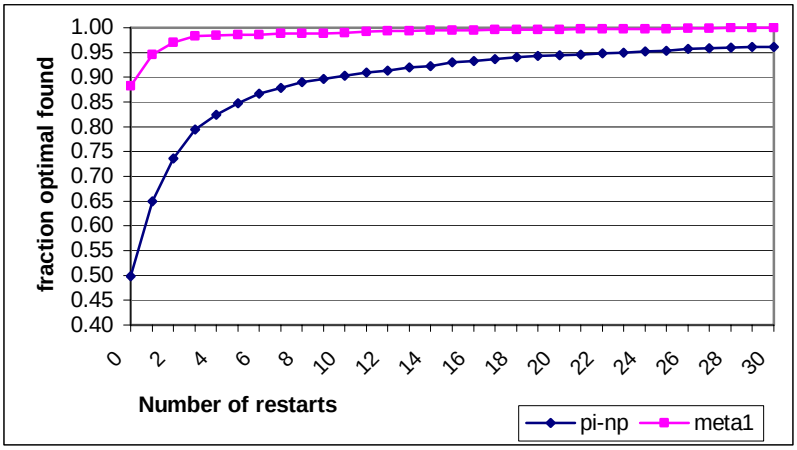


Figure 5. 18 Fraction Optimal Found for Problem P3

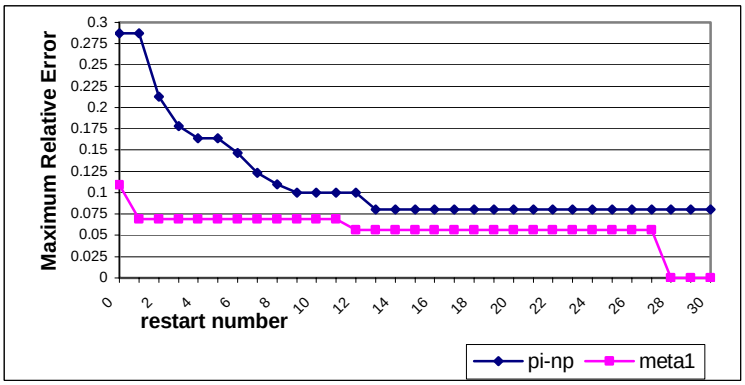


Figure 5.19 Maximum Relative Error for Problem P3

Figures 5.14, 5.16 and 5.18, shows after 30 random restarts, all of the problems are solved optimally. Although difficult to see in figure 5.18, 100% is not achieved until restart number 28, while others achieve 100% at less than 30. Also note that policy iteration without perturbation solves over 95% of the problem instances optimally, but higher restart numbers are required. Problem P3 has the longest duration in terms of restarts required for a good solution. To perform 30 random restarts takes on average 0.15 seconds, while $N/2$ (N equal to the number of states) takes on average 0.08 seconds. Although with 30 restarts all 1000 problem instances are solved, the execution time is longer than total enumeration. Recall, the number of states in this problem is $(C_s+1)*(C_r+1)$ where C_s and C_r are 4. There are a total of $(C_s+1)!$ possible solutions. To enumerate all solutions takes 0.10 seconds. Clearly for this problem, 30 random restarts are too much as enumeration can solve the problems optimally in less time. However, as

the problem size increases and enumeration becomes intractable, executing random restarts is faster. Table 5.2 illustrates the affect of executing *N* random restarts on larger problem instances. Note, there is no degradation in performance with random restarts. In the next section, we examine perturbations on the information vector and the effect of combining that strategy with the policy perturbation restart strategy.

Table 5. 2 Random Restart Results for Larger Problem Sizes

Problem size (C_s,C_r)	Fraction optimal found	Average relative error	Maximum relative error	Average Execution time (CPU Secs)
(5,5)	0.998	0.006	0.007	0.223
(6,6)	1	0	0	0.834
(7,7)	0.998	0.002	0.003	2.595
(8,8)	0.99	4.47955E-6	4.47955E-6	8.004

5.3 Sensitivity analysis with information vector perturbation

5.3.1 Overview

Based on the experiments from the previous chapters, strategy *pi*3 and *pi*4 yield the best results for the randomly generated problems, while *pi*2 yields the best results for the supply chain problem. The perturbation strategy based on adding a constant value to the information vector (*pi*1) is not effective and no longer considered a viable strategy. Perturbations based on generating neighbors for the current best iterate outperform the method of allowing movement away from the best iterate. Since the previous experiments only consider a fixed value of epsilon set to 1/*N*, where *N* is the number of states in the problem, we examine the sensitivity of the value of epsilon on performance. There may be other values of epsilon which provide better results and thus influence the choice of the dominating perturbation method.

The appropriate choice of termination criteria based on the problem size is also examined. The results of chapters 3 and 4 terminate after a fixed number of perturbations are executed. The objective is to find an appropriate value based on the size of the problem being solved.

The results from previous experiments indicate that perturbations using the information vector yield comparable results to perturbations based on the policy, at the expense of higher information vector perturbations. For example, in the (6,6) problem examined in section 3.3, it takes 40 information vector perturbations to achieve the same level of performance as local improvement based on policy perturbation. Further analysis studies the use of combining policy and information vector perturbations to achieve the best possible performance without compromising efficiency.

5.3.2 Sensitivity Analysis with epsilon

These experiments consider the supply chain problem with binomial retailer demand, (*s*,*S*) inventory control policy and supplier penalty cost of 3 (problem P1 in table 5.1). This is the same problem used above in the policy perturbation random restart analysis (figure 5.14). Additional problem sizes are considered to validate the choice of the best epsilon. Figure 5.20 displays the various perturbation strategies using different values of epsilon. Strategy *pi*2 is still the dominating perturbation method. From the perspective of strategy *pi*2, using smaller values of epsilon achieves the fastest increase in the least amount of perturbations. However, in the long run, the larger values of epsilon give the best result. When epsilon is equal to $1/N$, 88.2% are solved optimally after 60 perturbations, while 89.2% are solved optimally when epsilon is equal to $1/\sqrt{N}$. It is clear from this graph that using strategy *pi*2 is better for the supply chain problem. Since we are only considered with the supply chain problem, we will select that strategy for the information vector perturbations.

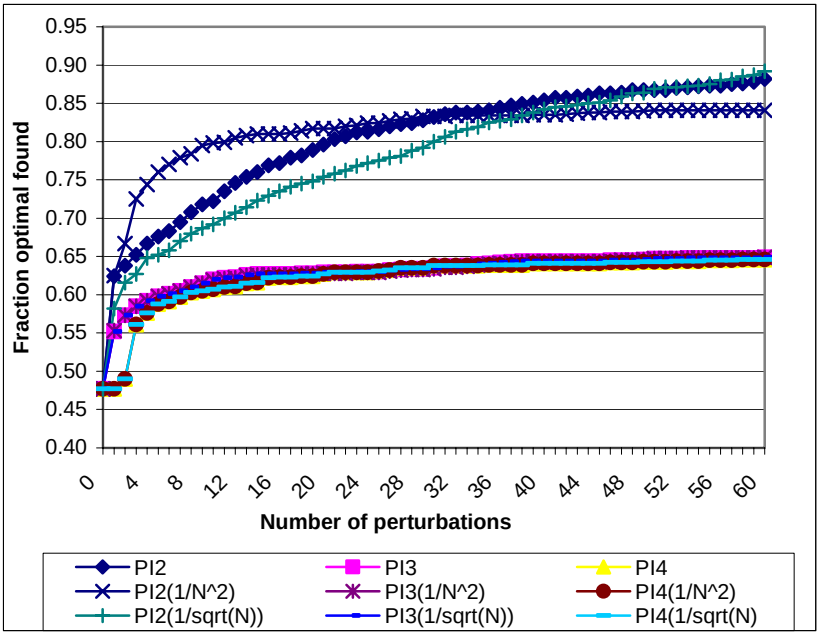


Figure 5. 20 Fraction Optimal Found -Epsilon Changing

Just focusing on strategy *pi*2, we show that for larger problem sizes, the larger values of epsilon continue to provide the best improvement, as shown in figures 5.19 – 5.21. Therefore, selecting $1/N$ or $1/\sqrt{N}$ for epsilon is suitable.

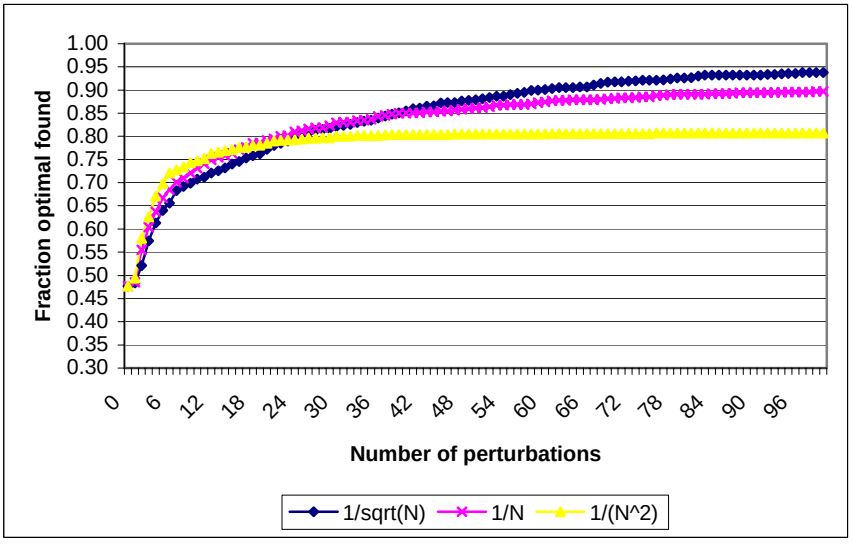


Figure 5. 21 Fraction Optimal Found for (4,4) Problem

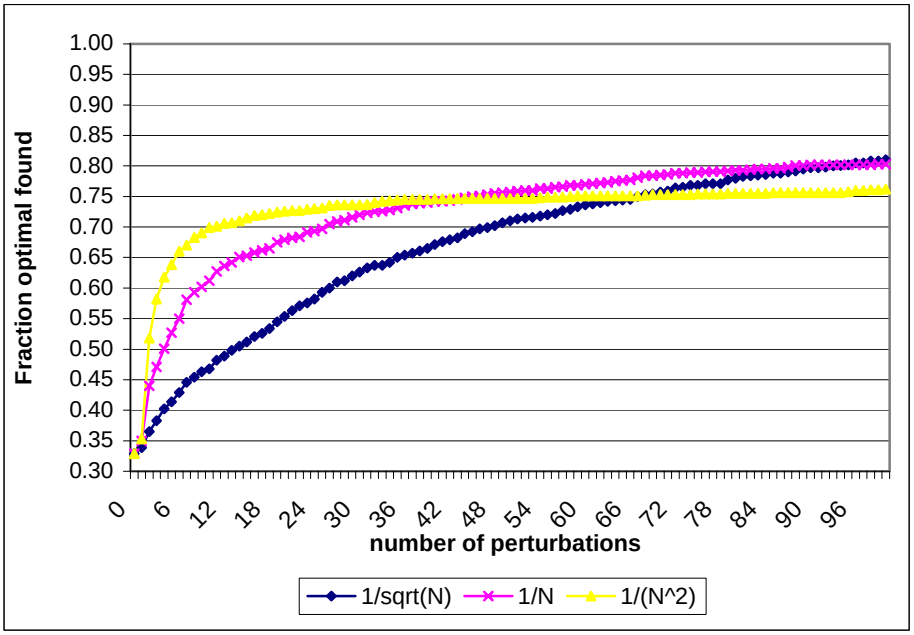


Figure 5. 22 Fraction Optimal Found for (5,5) Problem

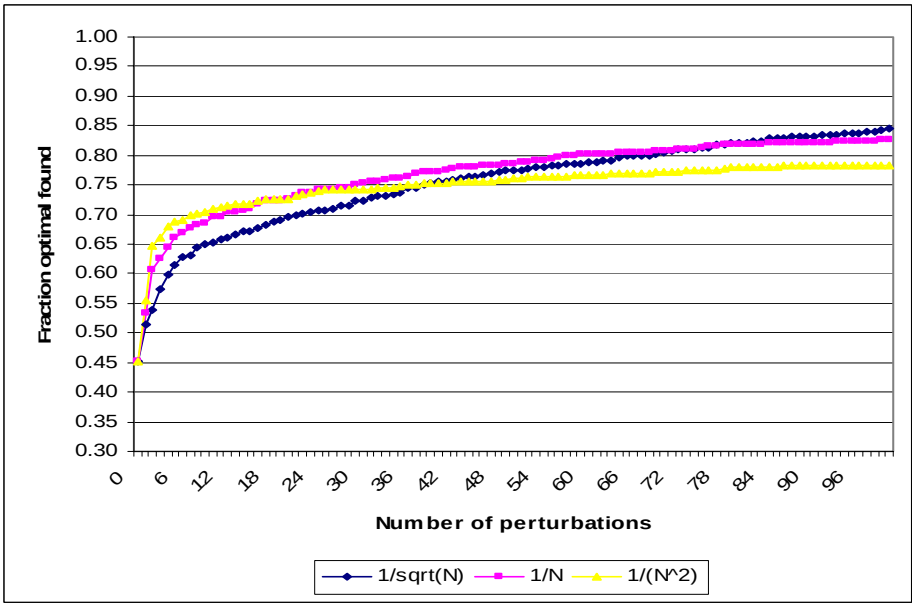


Figure 5. 23 Fraction Optimal Found (6,6) Problem

5.3.3 Termination Criteria based on problem size

Figures 5.21-5.23 show that even after 100 perturbations are performed, at least 20% of the remaining problems are still not solved optimally. The time to execute the various perturbation strategies is summarized in the table below. Performing

information vector perturbations is relatively cheap, but a large number are required to achieve good performance. For example, for the (4,4) problem it takes on average 0.014 seconds to perform 40 perturbations which solve 85.4% optimally. Recall that the policy perturbation strategy ls-*meta1* solved 85% in approximately 0.000661 seconds.

Table 5. 3 Information Vecotr Perturbation Performance

Problem size	Strategy	# of perturbations	Execution time	Fraction optimal found	Average relative error
(4,4)	*pi*2	40 = 1.6*N*	0.014	0.850	0.126
(4,4)	*pi*2	60 = 2.4*N*	0.020	0.874	0.126
(4,4)	*pi*2	100 = 4*N*	0.034	0.897	0.126
(5,5)	*pi*2	100 = 2.7*N*	0.069	0.810	0.048
(5,5)	*pi*2	60= 1.67N	0.043	0.736	0.056
(6,6)	*pi*2	100=2.1*N*	0.157	0.844	0.024
(6,6)	*pi*2	60=1.22*N*	0.102	0.788	0.031
(6,6)	*pi*2	4*N*	0.234	0.883	0.023
(6,6)	*pi*2	6*N*	0.446	0.921	0.028

Table 5. 4 Total Enumeration Execution Time

Problem size (C_s,C_r)	Total Enumeration (time in secs)
(2,2)	0.0002
(3,3)	0.0003
(4,4)	0.0074
(5,5)	0.1090
(6,6)	1.2838

For the (6,6) problem, after 100 perturbations on the information vector, 84.4% are solved optimally with average execution time of 0.157 seconds. For *meta1*, 87.5% are solved optimally using one iteration of policy perturbation with average execution time equal to 0.048 seconds. These examples illustrate that it takes more information vector perturbations to achieve the results of one iteration of policy perturbation. Random restarts with policy perturbations provide the best performance but are more expensive than iterations on the information vector. Since it is fairly 'cheap' (from a computation standpoint) to execute information vector perturbations, the performance of the algorithm can be improved by combining information vector perturbations with policy perturbations. The policy perturbation is performed first, followed by the information

vector perturbation. This constitutes one full iteration of the heuristic given some initial starting policy α. Table 5.5 summarizes the results when both policy and information vector perturbations are used together for supply chain problem P1. Recall for this supply chain problem, only 85.5% of the instances generated are solved optimally for the (4,4) problem size, 75.3% for the (5,5) problem size, and 75.9% for the (6,6) problem size.

Table 5. 5 Random Restart with Information Vector Perturbation Results for (6,6) Problem

Problem size (C_s,C_r)	**Epsilon**	**Random restarts**	**Information vector perturbations**	**Execution time (in seconds)**	**Fraction optimal found**	**Average relative Error**
(6,6)	1/*sqrt(N)*	*N*/5	2*N*	2.588	0.994	0.006
(6,6)	1/*sqrt(N)*	*N*/2	2*N*	6.571	0.997	0.004
(6,6)	1/*N*	*N*/5	2*N*	2.155	0.997	0.002
(6,6)	1/*N*	*N/2*	*N/2*	1.979	0.999	0.005
(6,6)	1/*N*	*N*/2	2*N*	6.571	0.999	0.001
(6,6)	1/(*N***N*)	*N*/2	2*N*	6.571	1.000	0

Table 5. 6 Random Restart with Information Vector Perturbation Results for (5,5) Problem

Problem size (C_s,C_r)	**Epsilon**	**Random restarts**	**Information vector perturbations**	**Execution time (in seconds)**	**Fraction optimal found**	**Average relative Error**
(5,5)	1/*N*	*N*/4	2*N*	0.695	0.998	0.004
(5,5)	1/*N*	*N*/2	2*N*	1.531	0.999	0.007
(5,5)	1/*N*	*N*/2	2*N*	1.531	0.999	0.007
(5,5)	1/*sqrt(N)*	*N*/2	2*N*	1.53	0.999	0.003
(5,5)	1/(*N***N*)	*N/2*	*N*	0.609	0.998	0.009
(5,5)	1/(*N***N*)	*N*/4	3*N*	0.647	0.995	0.009
(5,5)	1/(*N***N*)	*N*/2	2*N*	1.53	1.00	0

Table 5. 7 Random Restart with Information Vector Perturbation Results for (4,4) and (3,3) Problems

Problem size (C_s,C_r)	Epsilon	Random restarts	Information vector perturbations	Execution time (in seconds)	Fraction optimal found	Average relative Error
(4,4)	1/(*N***N*)	*N*/2	2*N*	0.300	0.999	0.007
(4,4)	1/*N*	*N*/2	2*N*	0.300	1	0
(4,4)	1/*sqrt(N)*	*N*/2	2*N*	0.300	1	0
(3,3)	1/*sqrt(N)*	*N*/2	2*N*	0.027	1	0
(3,3)	1/*N*	*N*/2	2*N*	0.027	1	0
(3,3)	1/(*N***N*)	*N*/2	2*N*	0.027	1	0
(3,3)	1/(*N***N*)	*N/2*	*N*	0.018	1	0

When using the combination of information vector perturbations with policy perturbations over 99.9% of the problems are solved optimally. Increasing the termination counter for the information vector perturbations to 2*N* achieves 100% solved optimally. The best results are achieved with the smaller values of epsilon on the (5,5) and (6,6) problem sizes. For the smaller problem sizes, comparable results are achieved regardless of the choice of the value of epsilon. Although not documented in the table above, the smaller value of epsilon for the (3,3) problem, solved all problem instances optimally in fewer iterations. Although the termination counter for information vector perturbations is set at 2*N*, the problems are solved optimally after two full iterations of policy and information vector perturbations when epsilon is set to 1/(*N***N*). When epsilon is set to 1/*N*, all are solved optimally after four iterations. All are solved optimally after five iterations when epsilon is set to 1/*sqrt*(*N*).

Table 5.8 displays the execution time in seconds for total enumeration (TE) and the random restart perturbation strategy (RR/PI). For the small problems, the execution time associated with *N*/2 restarts with 2*N* information vector perturbations exceeds total enumeration. However, total enumeration execution time increases exponentially as the problem size increases, while the random restart and perturbation strategy of the heuristic does not.

Table 5. 8 Execution Time for ROMDP Heuristic and Total Enumeration

Problem size (C_s,C_r)	TE (in secs)	RR/PI (in secs)
(2,2)	0.0002	0.0011
(3,3)	0.0003	0.0132
(4,4)	0.0074	0.0920
(5,5)	0.1090	0.3879
(6,6)	1.2838	1.6143
(7,7)	21.3479	5.5946

5.4 CONCLUSIONS

Incorporating random restarts into the ROMDP policy iteration/policy perturbation algorithm increases the percentage of finding optimal solutions when the structure of the solution space contains many local minima. One problem instance showed an increase of 89.7% when random restarts are used. Incorporating information vector perturbations into the random restart strategy can provide further improvement. As more information vector perturbations are employed, fewer random restarts are required. The results of table 5.5 illustrate the power of combining multiple strategies to tackle difficult problems. The structure of the algorithm using both random restarts and information vector perturbations is summarized below.

Step 0. Initialization

Generate an initial policy. Policy can be myopic or randomly generated. Initialize best gain g^* and best policy π^*.

Step 1. ROMDP Policy iteration/policy perturbation

Perform ROMDP policy iteration/policy perturbation as defined in section *3.2.2.4* to obtain best policy $\tilde{\pi}$ and associated gain $g^{\tilde{\pi}}$.

Step 2. Information Vector Perturbation

Perturb the information vector associated with policy $\tilde{\pi}$ as defined in section *3.2.2.5* to find a better policy and gain. Update $\tilde{\pi}$ and $g^{\tilde{\pi}}$ appropriately.

Step 3. Evaluate policy

If $g^{\tilde{\pi}} < g^*$ then set $g^* = g^{\tilde{\pi}}, \pi^* = \tilde{\pi}$

Step 4. Random restart/Termination Criteria

If the number of random restarts not exceeded, generate a new starting policy and return to step *1*.

Good results are achieved by setting the maximum number of information vector perturbations to be at least twice the number of the random restarts.

Chapter 6 Successive Approximation approach to ROMDP

6.1 Background

In order to analyze the information sharing problem for larger models, a more efficient procedure for determining the gain, relative values, and information vector associated with each policy is required. The algorithm constructed in chapter 3, relies on matrix inversion during the policy evaluation phase of the heuristic to obtain the associated policy measures. Considerable computational effort is required to obtain those values during the search for a local minimum. White (1963) introduces a more computationally efficient procedure by iterating on the policy improvement test quantity, also known as the value iteration equation. This method of successive approximations converges to the same unique solution determined by matrix inversion. The method of successive approximations introduced by White (1963) has also been studied by Odoni(1967), Hodgson and Koehler (1979), Hodgson and Zaldivar (1975), Morton (1971), Schweitzer *et al*. (1977), and Ding *et al*. (1988) for aperiodic markov chains. Su and Deninger (1972) provide a comparable procedure for periodic markov chains. The approach developed by Ding *et al*. (1988) applies to problems with a special structure, specifically, transition matrices that have a large number of transient states. Since many of the supply chain problems examined possess the properties necessary for this procedure, it will be used as the base for developing a successive approximation counterpart applicable to solving the ROMDP. The ROMDP procedure developed here applies to processes that are aperiodic and single chained. Treatment of periodic and multi-chain policies will be discussed in a later section.

6.2 Ding Procedure for undiscounted MDP

For an undiscounted, aperiodic, single chained markov process, White (1963) proves that repeated computation of the following value iteration equations will result in convergence to the optimal gain.

$$V_n(i) = \max_k \left[q_i^k + \sum_j p_{ij}^k v_{n-1}(j) \right]$$

$$g_n = V_n(m)$$

$$v_n(i) = V_n(i) - g_n$$

The subscript *m* denotes the index of a state in which there is a positive probability of returning after some sequence of decisions are made. Morton (1971) develops a parallel procedure, using the same value iteration equations defined above, which alternate between fixed policy successive approximation (cheap iterations) and policy maximization. This approach also results in convergence to the optimal gain and associated policy. Ding *et al*. (1988) develop an efficient procedure which takes advantage of the special structure of large scale MDPs to provide additional computation reduction. For large scale markov chains, Ding characterizes two key properties of the transition matrices which provide the foundation for the algorithm presented. He notes the transition matrices associated with large scale problems are frequently sparse, containing a large number of transient states. Thus, when testing if a given policy is optimal, the selection associated with a transient state does not influence the calculation of the relative values for the recurrent states or contribute to the calculation of the gain. He also notes that when there are a large number of transient states, the recurrent states of the optimal policy tend to cluster in a small number of compact groups. Taking these two observations into account, the Ding successive approximation procedure is as follows.

Step 1. Policy Evaluation

1. For a given policy, compute the limiting state probabilities to determine the set of recurrent states, *R*.
2. Find neighboring states within some radius *r* for all recurrent states. Let the set of all the neighboring states not in *R* be *A*.
3. Find all the states that are reachable (in one or more transitions) from *A* but not including the states in *A* or *R*. Let the set of these states be *C*.
4. Implement fixed policy successive approximation for the states in the set *R*+*A*+*C* for a fixed number of iterations. Formally,

$$y_i(n+1) = c_{ia} + \sum_j p_{ij}(a) w_j(n)$$
$$w_i(n+1) = y_i(n+1) - y_N(n+1)$$
$$i = 1,..., N$$

Step 2. Policy Improvement

5. Making use of the relative values w_i, calculated in step 4 for states in $R+A+C$, implement the policy improvement.

$$\max_{k \in K_i'} \left\{ c_{ik} + \sum_j p_{ij}^k w_j(n) \right\} i, j \in R + A + C$$

where K_i' denotes the set of alternatives for state i that can only make transitions to states in $R+A+C$.

Termination

6. If there is no change in the policy set and the $w_i(n)$'s have converged, stop. Otherwise go to step 1.

For problems with a large number of transient states, Ding *et al*. (1988) achieve significant computational reduction over the full state space successive approximation approach of Morton (1971).

6.3 Adaptation for ROMDP

6.3.1. Successive Approximation heuristic for ROMDP

In modifying Ding's (1988) procedure there are several considerations specific to the ROMDP which must be addressed. When determining the set of actions that can be used during policy improvement, the action evaluated can only make transitions into states contained in the set $R+A+C$. Recall the set of all alternatives for a given observation set S_k is defined as $A(k)$. If any action $a \in A(k)$ exists with $p_{ij} > 0$, $i \in R + A + C, i \in S_k$ and $j \notin R + A + C$, then that action must be discarded. This reduced set of admissible actions is denoted $A'(k)$. If during the policy improvement step, there are states for an observation set k in $R+A+C$ with no corresponding alternatives in $A'(k)$

then consider increasing the radius *r* used for calculating the neighboring states. Every new recurrent set R' is a subset of *R*+*A*+*C*. If the radius of the neighborhood is not significantly large enough to allow movement to a new recurrent set, the procedure can terminate on a suboptimal solution.

Ding *et al*. (1988) note that transient states do not effect the overall selection of the policy during policy improvement. However, to ensure that an improving policy is selected appropriately, several transient states (states in *A*+*C*) are included to complete the chain associated with the neighborhood of *R*. These states and corresponding relative values are used in the policy improvement step of the algorithm. In the ROMDP, the policy vector and policy improvement test quantity are functions of the observation set S_k. Formally, the new alternative for observation set S_k satisfies

$$\min_{a \in A(k)} \sum_{i \in S_k} x_i \left\{ c_{ia} + \sum_j p_{ij}(a) v_j \right\}$$

Further computational efficiency can be obtained for the ROMDP by performing the summation across all recurrent states $i \in S_k$. Since the limiting state probabilities associated with the transient states are effectively zero, they will not influence the selection of the policy for the observation set.

Termination defined in Ding's (1988) procedure occurs when there is no change in the policy set and the $w_i(n)$'s have converged. It is shown in chapter 3, that policy improvement for the ROMDP can lead to another policy with a higher gain (lower for maximization problem). When this occurs, the current solution which led to the non-improving policy is a local minimum. When evaluating a policy using simultaneous equations, we can determine the optimal gain associated with that policy and can therefore terminate on the current local minimum appropriately. However, in the successive approximation procedure, the optimal value of the gain is obtained only when the relative values have converged. Performing fixed policy successive approximation for some finite number of iterations results in an approximation of the gain if convergence of the relative values has not been achieved. If the termination criteria defined by Ding *et al*. (1988) were used for the ROMDP, it would be possible to cycle indefinitely among non-converged policies. Therefore, the approximation of the gain must be used to determine when a non-improving policy is found and convergence to a

local minimum is achieved. Odoni (1967) provides conditions for obtaining an upper and lower bound on the gain and proves the upper bound decreases monotonically and the lower bound increases monotonically as the number of iterations increase. The gain can be approximated as the average of the upper and lower bounds. This approximation is exact when the number of iterations is large (Odoni,1967). Making use of the Odoni bounds, convergence is achieved when the upper and lower bound are the same or within some value ε. The convergence based on these bounds is defined as the span semi-norm and can be calculated very easily during the value iteration step (step 4) of Ding's procedure. In addition, implementing these bounds in the context of the ROMDP provides a simple mechanism for identifying suboptimal policies during the local improvement phase. Implementing the modifications discussed above ensures termination can occur on a local minimum in the same manner as described in chapter 3.

Making use of the observations above, a successive approximation counterpart to the policy iteration heuristic defined in 3.2.2 is given below. The notation is in line with the notation defined in Chapter 3. Items superscripted with * denote converged values and the associated policy. Items superscripted with ~ denote approximate (non-converged) values and the associated policy.

Step 0. Initialization

Generate an initial admissible policy, π.

Set $g^* = \infty$

Set $(R+A+C) = S$

Step 1. Policy Evaluation

a. For a given policy, π, compute the limiting state probabilities in order to determine the set of recurrent states, R_π. That is, for a fixed number of iterations, compute

$$x_i(n+1) = \sum_{j \in R+A+C} x_j(n)\, p_{ji}\left(\pi_{G(j)}\right)$$

b. Find neighboring states within some radius *r* for all recurrent states. Let the set of all the neighboring states not in R_π be A_π.

c. Find all the states that are reachable, in one transition, from A_π, which are not already in $(R+A)_\pi$. Denote these sets of states as C_π. Formally, find all states

$j \notin (R+A)_\pi$ *where* $p_{ij}(a) > 0 \; \forall i \in S, a \in A$

d. Implement fixed policy successive approximation for the states $i \in (R+A+C)_\pi$ for a fixed number of iterations using the value iteration equation defined below. Compute the Odoni (1969) upper (L'') and lower (L') bounds on the gain (g^π) as follows.

$$y_i(n) = c_{i\pi_{G(i)}} + \sum_j p_{ij}(\pi_{G(i)}) w_j(n) \quad i = 1,...,N$$

$$w_i(n+1) = y_i(n) - y_N(n) \quad i = 1,...,N$$

$$L'_n(\pi) = \min_i (y_i(n) - w_i(n))$$

$$L''_n(\pi) = \max_i (y_i(n) - w_i(n))$$

$$g^\pi = \frac{L'_n(\pi) + L''_n(\pi)}{2}$$

Step 2. Bounding and Convergence test

<u>*Case a*</u>: $\text{span}(w^{n+1} - w^n) = L''_n(\pi) - L'_n(\pi) < \varepsilon$

If $g^\pi < g^*$ set $g^* = g^\pi, \pi^* = \pi$ and proceed to step 3. Otherwise, the current policy π^* with gain g^* is a local minimum. Proceed to local improvement (policy perturbation) phase.

<u>*Case b*</u>: $\text{span}(w^{n+1} - w^n) = L''(n) - L'(n) > \varepsilon$.

If $g^\pi < \tilde{g}$, set $\tilde{\pi} = \pi, \tilde{g} = g^\pi$ and proceed to step 3.

If $g^\pi > \tilde{g}$, the current best non-converged policy $\tilde{\pi}$ may be a local minimum. Repeat step *1d* for states $i \in (R+A+C)_{\tilde{\pi}}$ until convergence is achieved or $L'(\tilde{\pi}) > g^*$. If g^* exists ($g^* <> \infty$) and $\tilde{g} < g^*$, then $\tilde{\pi}$ is a local minimum. Update π^* and associated gain g^* appropriately. Terminate with the current policy π^* with gain g^* as a local minimum. Proceed to local improvement phase.

Step 3. Policy Improvement

e. Making use of the relative values w_i calculated in step *1d*, implement the policy improvement for all observation sets $k \in O$ with action space $A'(k)$ (defined earlier) and proceed to step 1.

$$a.\quad \pi_k = \min_{a \in \breve{A}(k)} \sum_{i \in R_k} x_i \left(c_{ia} + \sum_{j \in (R+A+C)_\pi} p_{ij}(a) v_j \right)$$

where R_k denotes the set of recurrent states for observation set S_k.

The bounding and convergence step ensures that time is not wasted forcing a suboptimal policy to convergence. If a current solution has been obtained but has not yet converged, then computational effort is expended toward the convergence of the identified local minimum. Recall in chapter 3, a local minimum is reached when no improving feasible direction (via policy improvement) exists. That is, the information vector and relative values associated with a policy π lead to either the same policy or a new policy with worse gain. This converged gain and associated policy serves as the starting point for the local improvement procedure. When performing the local improvement phase of the heuristic, simply substitute the evaluation steps define in 3.2.2.4 with the successive approximation approach defined above. When a neighboring policy is evaluated, perform fixed policy successive approximations on that neighbor until one of two things occur: The lower bound on the gain exceeds the best gain obtained thus far and the neighbor can be eliminated; Convergence to an improving gain is achieved after some fixed number of iterations. If a better neighboring solution is found, then policy iteration is restarted as described in chapter 3.

6.3.2 Periodic and Multi-Chain policies

Since the ROMDP recurrent state encapsulation method is based on White's (1963) method of successive approximation, which converges for a completely ergodic process, then any policy in the ROMDP that leads to a completely ergodic process will also converge as long as an acceptable recurrent state *j* is chosen such that $p_{ij}^{k_1 k_2 \ldots k_{u+1}} \geq \alpha$ for some α in [0,1). However, in the ROMDP, there may be policies that are constructed during the course of policy improvement which induce markov chains which are periodic or multi-chain. In this case, a limiting distribution does not exist (in the sense of White). This is potentially problematic if the radius for calculating neighboring states is not large enough to capture all of the states in the single chain. During calculation of the information vector you may capture the recurrent states

associated with a certain period, which can lead to unexpected or erroneous results. If the periodic chains can be detected during the course of determining the recurrent state set, then the associated policy can be discarded or the underlying transition matrix transformed to an aperiodic process. Putterman (1994) provides transformation equations for converting a periodic transition matrix into an aperiodic one. The transformed MDP has components defined as follows:

$$\tilde{c}_{ia} = \tau c_{ia} \quad a \in A, i \in S$$
$$\tilde{p}_{ij}(a) = (1-\tau)\delta(j \mid i) + \tau p_{ij}(a)$$
$$\delta(j \mid i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Under this transformation, the average optimal stationary policies for the original and transformed problem are identical (Putterman 1994). The gain of the original model is proportional to the gain of the transformed model with proportionality factor $1/\tau$. During the computation of the gain and relative values, it is not necessary to store or create a new matrix representing the transformed transition probabilities and reward structure. Simply store the functional relationship and use the transformed values in the successive approximation computations. The interpretation of this model under a discrete time system is that at each decision point, the system remains in the current state with probability τ regardless of the decision chosen. Performing this transformation allows the solution associated with the policy to be solved in the same manner that it is solved under the simultaneous equations method. For a single chain periodic process, a stationary distribution is found via the same set of simultaneous equations used for calculating the limiting distribution for a single chained aperiodic process. Although the process is periodic, the underlying matrix for determining the gain, relative values, and associated information vector is invertible. Su and Deninger (1972) provide a successive approximation approach to address periodic policies. In this research, we are only concerned with single-chained, aperiodic processes. If a periodic policy is chosen, we can apply the transformation described above and move to a new policy via policy improvement. The detection of a periodic process can introduce additional computation time into the heuristic. During the course of the iterative procedure for determining the limiting state probabilities, the index associated with recurrent state with the largest

probability is saved. All relative values are scaled during value iteration using this recurrent state. For an aperiodic process, this recurrent state index converges to a single value as the limiting state probabilities for each state in $R+A+C$ converges. However, if the process is periodic, this index may alternate between two or more state values, depending on the length of the period. This is simple to detect for a cycle of length two. Anything longer may be computationally wasteful. An alternative approach is to assume if convergence has not occurred after some fixed number of iterations, then the policy is periodic. Careful consideration to the iteration number must be given so that well behaved policies are not discarded. If it is suspected that a problem under study contains many periodic policies, it is best to consistently apply the periodic transformation during computation rather than apply the detection procedures above.

Detection of multi-chain policies is considerably easier based on the work done by Fox and Landi (1968). They developed a computationally efficient labeling algorithm for determining the number of ergodic chains in a Markov process. Their algorithm has computational complexity of $O(|S^2|)$, where $|S|$ denotes the number of states in the process. If a new policy is selected via policy improvement, the Fox and Landi (1968) algorithm can be used to determine if the underlying transition matrix is single-chained. If it isn't, then the policy can be discarded. Incorporating these detection procedures within the context of the ROMDP algorithm is fairly straightforward. During the policy iteration phase, if a converged policy has already been obtained, discard the multi-chain policy and terminate on the converged solution. During policy perturbation, if the neighboring policy is multi-chain, then discard the neighbor as infeasible. A similar approach can be taken for periodic policies as an alternative to computing the transformed values.

6.4 Experimentation

The experiments aim to show the efficiency, with respect to computation time, and the effectiveness (with respect to percent solved) of the successive approximation approach to the ROMDP. The impact of the radius on both the computation time and quality of the solution has already been studied by Ding *et al*. (1988) and thus not considered here. It is sufficient to state only that as the radius increases, the computation

time increases and approaches the execution time of the Morton (1971) successive approximation procedure. The focus of this experimentation is on showing that good results can be obtained once the correct radius, via experimentation, is chosen. In the experiments conducted by Ding *et al*. (1988) a radius of two provided good results for most of the problems solved. Therefore, these experiments will start with that value and increase as necessary.

We first consider the supply chain problem introduced in chapter 5 with randomly generated discrete distribution representing the demand experienced by the retailer. The retailer employs an order-up-to replenishment policy. The supply chain problem with binomial demand, and an (*s*,*S*) retailer inventory control policy (*s* equal to $C_r/2$ and *S* equal to C_r) is also considered. The latter problem is already shown in chapter 5 to be difficult to solve and provides the worse performance without random restarts. Therefore, this is a good problem for considering the effectiveness of the successive approximation heuristic. A total of 10 cheap iterations are used per policy evaluation. A maximum of 180 total iterations are allowed for determining the converged gain associated with a policy (step 2*b*).

6.4.1 Performance with respect to optimal solutions

Table 6.1 displays the fraction of instances solved optimally using successive approximations approach. For each problem size, 1000 instances are generated with the exception of the (8,8) case. In this case, only 100 problems are generated due to the computation time required to enumerate all possible solutions. The problems generated reflect the supply chain problem with discrete demand that is randomly generated and retailer order-up-to policy. All of the problems are solved optimally with the exception of the (5,5) case. However, the one problem not solved is very close to optimal as indicated by the maximum relative error. A total of *N* random restarts and 0 information vector perturbations are used.

Table 6. 1 Results for Randomized Discrete Distribution and Base Stock (C_r) Policy

(Cs,Cr)	Policy Space	Fraction Optimal found	Average relative error	Maximum relative error	Execution time
(3,3)	18	1	0	0	0.236
(4,4)	96	1	0	0	0.364
(5,5)	600	0.999	0.007	0.007	0.568
(6,6)	4320	1	0	0	1.505
(7,7)	35280	1	0	0	3.625
(8,8)	322560	1	0	0	9.048

Table 6.2 displays the results associated with the supply chain problem reflecting binomial external demand and (*s*,*S*) retailer policy. Again a radius of 2 and total of *N* random restarts with 0 information vector perturbations are used.

Table 6. 2 Results for Binomial Demand Distribution and (s,S) Retailer Policy

(Cs,Cr)	Policy Space	Fraction Optimal found	Average relative error	Maximum relative error	Execution time
(4,4)	96	1	0	0	0.415
(5,5)	600	0.998	0.002	0.001	0.838
(6,6)	4320	0.999	0.009	0.009	2.324
(7,7)	35280	0.981	0.004	0.009	5.497
(8,8)	322560	1	0	0	13.828

For the (7,7) problem, a radius of 4 is used, which results in 21 problems not being solved optimally. However, the average relative error is 0.004 and the maximum relative error of 0.009. So, the solution found is very close to optimal. This example illustrates the importance of selecting the proper radius for the problem under investigation. There are three control parameters driving the effectiveness of the heuristic; the radius, the number of random restarts, and the maximum number of iterations allowed for policy convergence.

If the radius for determining neighboring states is too small, one or more improving policies will be missed and thus the algorithm will terminate on a suboptimal policy. This situation is observed in the (7,7) problem instances of table 6. Ding *et al*. (1988) also noted that their approach may terminate on a suboptimal solution if the radius is not large enough to capture the states associated with the optimal chain. To mitigate this result, a simple extension to the procedure is made by checking a larger

radius on the last policy iteration after convergence is achieved. This extension can be easily incorporated into the ROMDP by doing an additional iteration of the policy iteration heuristic at a larger radius once all randomly generated starting points have been exhausted. The best solution obtained from the random restarts is used as the starting policy for policy iteration.

Clearly, if the problems under study take longer to converge than the maximum number of iterations allowed, then the algorithm will never find a converged policy via policy iteration or policy perturbation. When a local minimum policy has been determined via the conditions defined in the bounding and convergence test, if the gain associated with that policy has not converged (within some ε-value), then the equations of step 1*d* must be performed until convergence is achieved or some maximum number of iterations is exceeded. Most policies converge quickly and termination occurs once the span semi-norm is less than ε. However, for policies not meeting the convergence criteria and exceeding the maximum iterations, we are left with a solution of which we can not determine the converged value of the gain. In the case of random restarts, we just start again with another policy hoping that the policy iteration phase will terminate on a local minimum which can be used to start the perturbation phase. There is no way to know if the discarded policy is indeed the optimal policy. If the maximum allowed number of iterations is too low, that policy may never be found. The maximum allowed iteration must be chosen carefully as to not discard potentially improving or optimal policies. In this experiment, instances did occur in which the policy iteration phase terminated on a non-converged policy. However the experimental results indicate that random restarts and the perturbation component aide in overcoming that problem by allowing for better starting points and ultimately better policies to be found in the search for a global minimum.

The selection of the number of restarts in previous chapters has been based on the size of the state space, *N*. As the problem size increases, one would believe that less restarts should be needed since more policy perturbations are being performed. Recall from chapter 5, the number of policy perturbations is

$$\binom{C_s}{2} + C_s$$

which, as a function of the number of state groupings (C_s+1), is on the order of $O\left((C_s+1)^2\right)$. In the successive approximation case, the actual number can be less since the number of states restricts the number of changes that can be made during policy improvement. If a state grouping contains no states in the set *R+A+C*, then the associated alternative is not changed. Therefore, the number of restarts required for a good solution should decay as the size of the problem increases. This is observed in experimentation when larger state spaces (400 plus states) are used. A simple check can be added every 100 restarts to see if the solution obtained is getting better. If it is, continue with the restarts, otherwise terminate.

6.4.2 Performance with respect to computation time

Table 6.3 summarizes the computation time for the ROMDP based on size of the policy space enumerated. The execution time in CPU seconds is shown for total enumeration (TE), successive approximation (SA), and simultaneous equation (SE). Comparison of the successive approximation time and simultaneous equation time is done without the use of restarts. In addition, the successive approximation approach uses a fixed radius of 2. The successive approximation procedure is more efficient then simultaneous equations after the (10,10) problem instance. In addition, table 6.4 reflects the procedure is efficient for larger problem sizes as well.

Table 6. 3 Execution time in CPU Seconds

(Cs,Cr)	Policy Space	TE time	SE Time	SA time
(3,3)	18	0.0003	0.0009	0.00447053
(4,4)	96	0.0074	0.002	0.007543
(5,5)	600	0.1090	0.006	0.01503
(6,6)	4320	1.2838	0.0164	0.0304675
(7,7)	35280	21.3479	0.039	0.052214
(8,8)	322560	1022.750	0.250	0.110156
(10,10)	36288000	-	0.366	0.358833
(15,15)	1.962E+13	-	6.314	3.2955
(20,20)	4.866E+19	-	127.886	12.2835

Table 6. 4 Execution Time in CPU Seconds for Larger State Spaces

(Cs,Cr)	Number of States	SA time
(10,10)	121	0.359
(20,20)	441	12.284
(30,30)	961	61.993
(40,40)	1681	235.549
(50,50)	2601	729.865

Chapter 7 A Case for Information Sharing

7.1 Problem Description

In the previous chapters, considerable attention was given to showing that the ROMDP algorithm provides good results for solving the no information sharing problem. That algorithm, along with Howard's (1960) policy iteration heuristic is used to study the value of information sharing in a two-stage supply chain. Recall, a completely observable MDP denotes a model with full information sharing. Consider a two stage supply chain structure consisting of a supplier and a retailer. The supplier replenishes his inventory from an exogenous source that has infinite capacity. The supplier is the single source used by the retailer to meet its customer demand. The retailer implements a fixed inventory control policy. The type of policy is discussed in the experimentation. The delivery lead time is assumed to be one period and therefore, orders placed at the beginning of the period are received at the end of the period. The sequence of events during a period is as follows.

1. The retailer examines his inventory and places an order.
2. The supplier receives the order and ships the available quantity from inventory. Any portion not filled from inventory is lost.
3. Supplier makes his production decision according to the decision policy.
4. Retailer demand occurs. Excess demand is lost.
5. Costs are calculated.
6. The retailer's order quantity is received into inventory.
7. The supplier's production quantity is received into inventory.

The cost model in the experiment represents the expected supply chain costs incurred during the period.

$$G_{sc} = h_s(i - z_r)^+ + p_s(z_r - i)^+ + [(h_s + h_r)(j - d)^+ + p_r(d - j)^+]p_D(d)$$

Measures used to quantify the value of information sharing (VOI) are the long run average cost (gain), long run average inventory level at the supply chain partners and the long run average lost sales incurred by the retailer. Let VOI_r represent the relative cost

reduction associated with the value of information sharing and $\phi(\alpha)$ denote the gain associated with policy α. Then VOI_r can be represented as

$$VOI_r = \frac{\phi(\alpha^*_{nis}) - \phi(\alpha^*_{is})}{\phi(\alpha^*_{nis})}.$$

Let $\bar{I}_m$ and $\bar{L}_m$ denote the long run average inventory level and lost sales of supply chain member *m*. x_i denotes the steady state probability of being in state *i*. Recall with no information sharing, the set of observable outputs (*O*) represents the supplier's inventory level. Each observation set S_k partitions the state space based on the supplier's inventory level $k \in O$. Therefore, the long run average inventory level of the supplier can be determined as

$$\bar{I}_s = \sum_{k \in O} k\left(\sum_{i \in S_k} x_i\right)$$

Each state *i* in the state space defines a two dimensional variable representing the supplier and retailer's inventory position. Therefore, a similar representation of the state partitioning can be made based on the retailer's inventory position and the corresponding average inventory level determined. Let *d* denote the demand observed during the period with probability p_d. Let *j* represent the retailer's current inventory position. Assume we have partitioned the retailer states according to the method described above and denoted that partitioning as R_k. Each observation set R_k partitions the state space based on the retailer's inventory level $k \in O_r$. Then the long run average lost sales can be determined as

$$\bar{L}_r = \sum_{k \in O_r} \sum_{j \in R_k} \left(\sum_d (d-k)^+ p_d x_j\right)$$

The supplier's average lost sales can be determined in the similar manner from the retailer's order quantity, z_r. In addition to the measures for evaluating the value of information sharing, we are interested in studying the influence that demand, supplier capacity, retailer policy and cost have on the value of information sharing.

The two stage supply chain model has been studied by Gavirneni *et al*. (1999), Lee *et al*. (2001), Yu *et al*. (2001) and Zhao and Simchi-Levi (2002). Lee *et al*. (2001) and Yu *et al*. (2001) consider uncapacitated models, while Gavirneni *et al*. (1999) and Zhao and Simchi-Levi (2002) consider capacitated suppliers only. We consider capacity constraints for both the supplier and the retailer. All previous papers consider excess

demand at the retailer to be backlogged, while we consider excess demand to be lost. Another common assumption among the previous models is that the retailer always gets all units demanded, regardless if the supplier can fill it or not. The missing part of the order that can not be filled from the supplier's inventory is usually assumed to come from some outside source that has to be restocked in the next period. In that respect, the supplier orders are also backlogged. We consider orders to be lost at both stages in the supply chain. Yu *et al*. (2001) study the value of information sharing on the total supply chain costs while Gavirneni *et al*. (1999), Lee *et al*. (2001), and Zhao and Simchi-Levi I (2002) consider the value of information as a function of the supplier's cost and thus no benefit to the retailer is quantified.

7.2 Demand effect on value of information sharing

7.2.1 Design of experiment

First, we consider the impact of demand on the value of information sharing. Assuming the supplier and retailer capacity are fixed at 20 units, three demand distributions are considered: Binomial; Discrete Uniform; and Poisson. The maximum demand that can occur in the period will not exceed the retailer's capacity. The mean demand is the same for each distribution. For a binomial distribution, the mean (μ) and variance (σ^2) are functions of the success probability *p*. Formally, $\mu = np$ and $\sigma^2 = np(1-p)$. For Poisson, the mean and variance are the rate of event occurrence per unit time (λ), namely the demand occurring per period. For the discrete uniform, the mean and variance are functions of the minimum and maximum demand interval limits [*a*,*b*]. The mean is defined as the average of the interval limits and the variance is defined by the following equation.

$$\sigma^2 = \frac{(b-a+1)^2 - 1}{12}$$

Let $\bar{c}$ denote the vector of holding and penalty costs for the supply chain problem with components (h_s, p_s, h_r, p_r). For this experiment, $\bar{c} = (1,3,1,14)$. Mean values of demand ranging from 12 to 18 are considered. In all cases, the ratio of capacity to mean demand

is greater than one. Additional analysis where capacity is less than mean demand will be discussed in section 7.3.

In order to study the effect of the variance on the value of information sharing, we generate problem instances using a discretized normal distribution as well as the discrete uniform. With the discretized normal distribution we can keep the mean and range of demand values constant while changing the value of the variance. However, with the discrete uniform distribution, in order to keep the mean constant the range of possible demand values must be changed in order to generate varying values of the variance.

7.2.2 Results

Table 7.1 summarizes the distribution parameters for the experiments performed.

Table 7. 1 Distribution Parameters

Distribution	$\mu = 8$	$\mu = 10$	$\mu = 12$	$\mu = 15$
Binomial	p = 0.6	p = 0.5	p = 0.6	p = 0.75
Poisson	$\lambda = 8$	$\lambda = 10$	$\lambda = 12$	$\lambda = 15$
Uniform	[0,16]	[0,20]	[4,20]	[10,20]

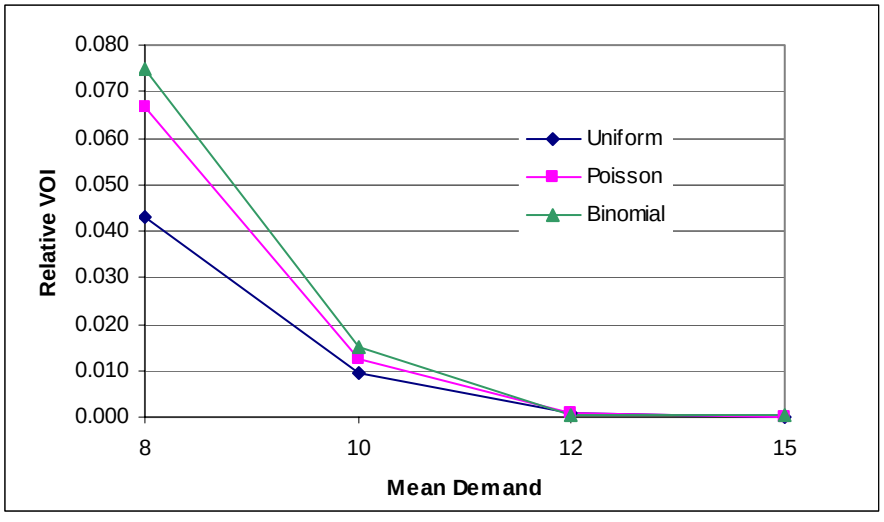


Figure 7. 1 Relative VOI versus Demand

Figure 7.1 displays the relative value of information sharing as a function of mean demand. When the mean demand is 10, the value of information sharing is highest for the distribution with the lowest variance (binomial distribution) and decreases as the demand variance increases. As the mean demand increases and approaches the retailer capacity, the relative value of information sharing is nearly identical for all distributions and approaches zero. Since we are modeling a capacitated supply chain structure, the value of information sharing is affected by the excess or lack of excess capacity available to make production decisions. Section 7.2 discusses the capacity affect on the value of information sharing in detail. Here, we mainly focus on the demand distribution and demand variance.

When the mean is held constant, figure 7.2 shows the value of information sharing increases then decreases. When the variance and resulting coefficient of variation is small, the value of information increases. As the variance increases, the value of information sharing starts to decrease. Increasing the variance increases the range of possible demand values and pushes the system closer to the capacity limit. This suggests that at higher demand variances, little improvement in the system can be achieved with sharing information when both supply chain members are capacitated. Overall, the value of information sharing is relatively small, not exceeding more than 1.6%.

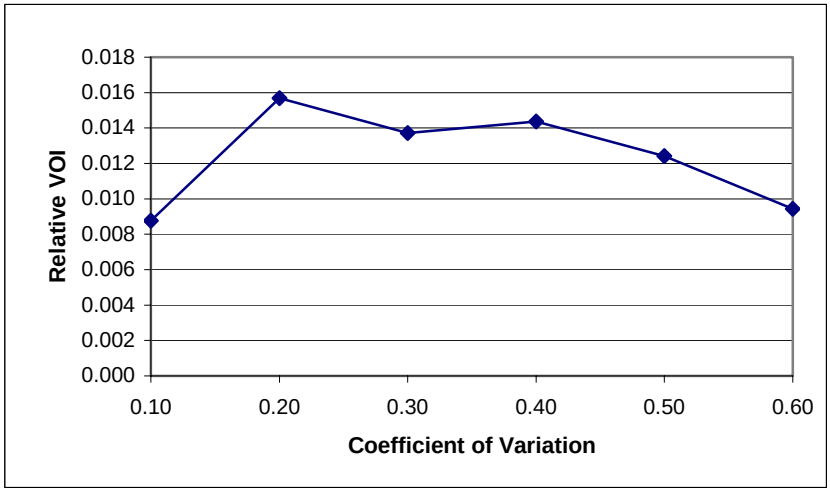


Figure 7. 2 Coefficient of Variation Effect on VOI for Discretized Normal Distribution (Mean 10)

7.3 Capacity effect on value of information sharing

7.3.1 Design of Experiment

Next, we study the effect of the supplier capacity on the value of information sharing. The mean demand incurred by the retailer is fixed and the capacity of the supplier is a function of the mean demand (μ_r). We match the means of the three distributions and vary supplier capacity with respect to mean demand. Instances where the supplier capacity is $0.80\,\mu_r$, $0.90\,\mu_r$, $1.0\,\mu_r$ $1.1\,\mu_r$ and $1.2\,\mu_r$ are considered. Three distributions examined are summarized in table 7.2 with the associated parameters. An order-up-to (base stock level C_r) replenishment policy is used by the retailer.

Table 7. 2 Distribution Parameters for Capacity Analysis

Distribution	C_s=13	C_s=14	C_s=15	C_s=17	C_s=20
Binomial	p = 0.75	p = 0.75	p = 0.75	p = 0.75	p = 0.75
Poisson	$\lambda = 15$	$\lambda = 15$	$\lambda = 15$	$\lambda = 15$	$\lambda = 15$
Uniform	[10,20]	[10,20]	[10,20]	[10,20]	[10,20]

7.3.2 Results

7.3.2.1 Value of information sharing

Figure 7.3 shows that significant value can be achieved at lower capacities when information concerning the retailer's inventory position and demand is shared. However, as the capacity of the supplier increases, the value associated with the information decreases, approaching zero in the case when the supplier and retailer's capacity are identical. When supplier capacity is significantly less than mean demand there is little value in sharing the information. The supplier's only option is to produce to maximum capacity at all times. As additional capacity becomes available, the relative VOI starts to increase. The supplier can scale back production in certain states to mitigate costs of holding excess inventory while still satisfying the retailer's replenishment orders. The maximum value associated with sharing information is achieved when supplier capacity is half of the retailer's capacity, as illustrated in figure 7.3. From that point, any increase in supplier capacity results in a decrease in the relative VOI decreases. This decrease is

attributable to the myopic policy under no information sharing improving. The information sharing policy is not changing and therefore the lowest possible cost in the system can be achieved at a lower level of capacity (67%).

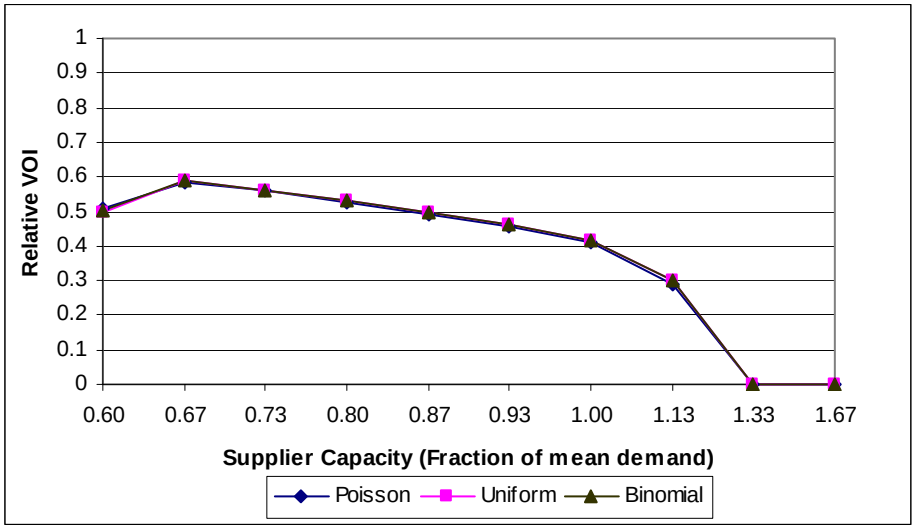


Figure 7. 3 Relative VOI when mean demand = 15

From an overall reduction in lost sales, figure 7.4 shows the supplier receives the most benefit achieving 100% reduction. This is obvious as the increase in system inventory helps to offset the lost sales incurred by both parties. Without information sharing, the echelon inventory position in the system is lower because the supplier's decisions are based solely on his inventory position driven by the retailer's replenishment orders. When information is available prior to making the production decision, the supplier's production quantity is higher. He can build up inventory and better satisfy the retailer's requests.

A portion of the expected cost savings achieved by the supplier under a full information sharing policy can be passed to the retailer as an incentive for sharing his demand and inventory information. The amount the supplier is willing to pay to the retailer for obtaining real time access to the retailer's information is a function of VOI_r.

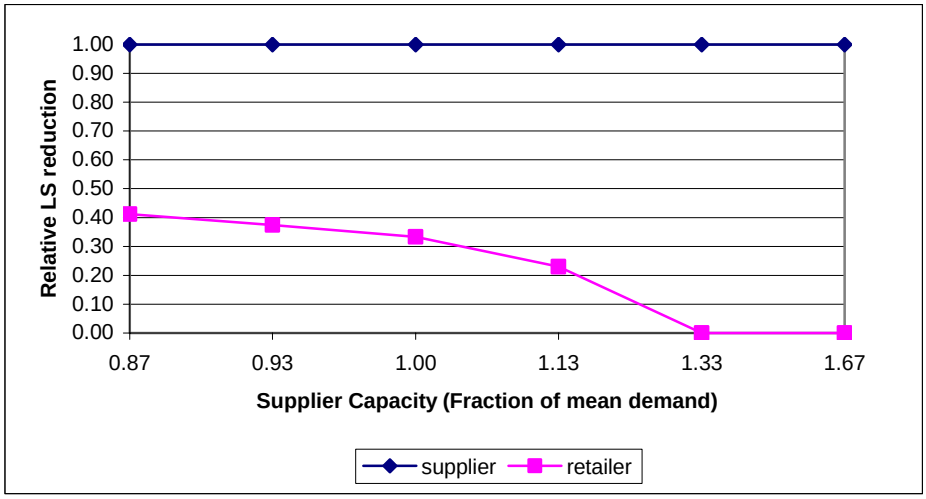


Figure 7. 4 Relative Lost Sales Reduction with Information Sharing

From a retailer perspective, it may seem the supplier gets most of the benefit. However, there is a direct relationship between the supplier's production capability and the retailer's available inventory. Figure 7.5 shows the expected production output as a function of capacity for binomial demand. The other demand distributions have the same form and are therefore not shown. The production output of the supplier is exactly what is available for the retailer to meet their demand requirements. As production output increases, the retailer's lost sales decrease. This relationship is depicted in figure 7.6 for the policy when no information is shared. If the supplier is operating significantly below the mean retailer demand, the supplier can satisfy replenishment orders at a greater rate per period once the retailer's inventory position and incoming demand are shared. As illustrated in figure 7.3, as capacity increases and approaches the retailer's capacity level there is little gained in sharing information.

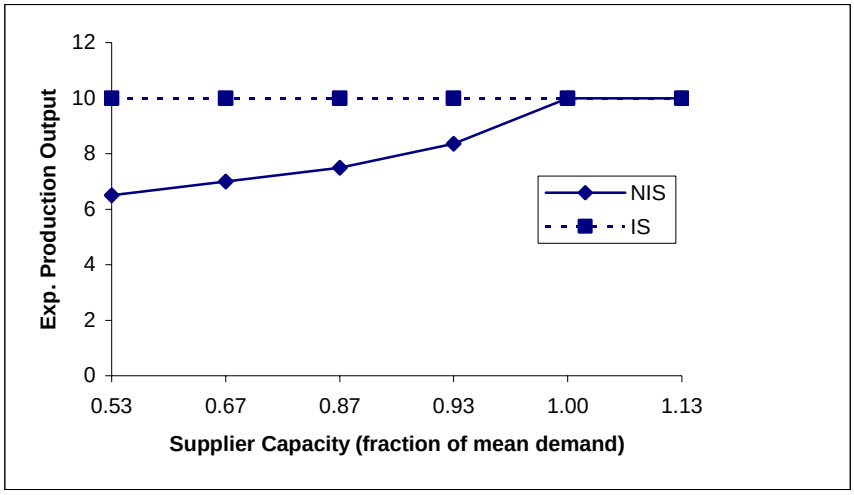


Figure 7. 5 Expected Supplier Production Output for Binomial Demand Problem

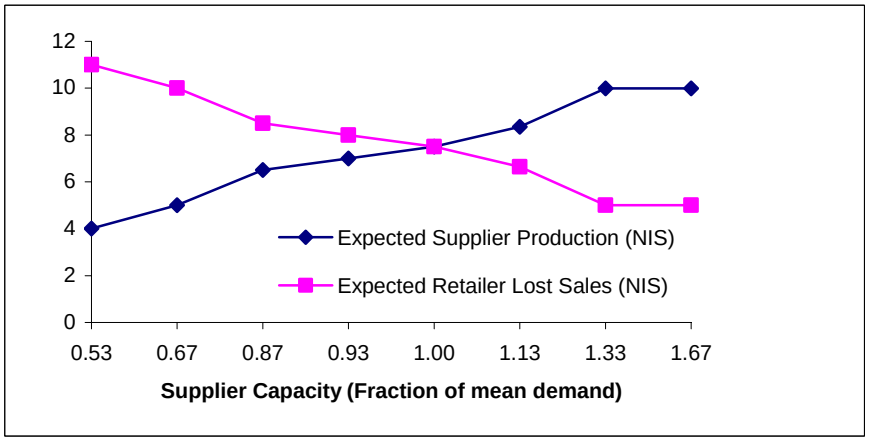


Figure 7. 6 Expected Retailer Lost Sales and Supplier Production for Binomial Demand Problem

7.3.2.2 Optimal production control policy under steady state

In the absence of information, the supplier operates as close as possible to an (*s*,*S*) policy, delaying production until his inventory position falls below *s*. When $s = S$, the policy is an order up to policy. Figure 7.7 shows the value of *s* as a function of the supplier capacity. In some instances, the value of *s* is 1. The supplier produces to capacity in one period and then is idle in the next, as his only recurrent states are C_s and 0. When information is not shared, the supplier in effect is making production decisions at the beginning of the period. If his current inventory position is equal to his maximum capacity, then he will not produce. With this type of policy, the supplier is utilizing his production resource about 50% of the time.

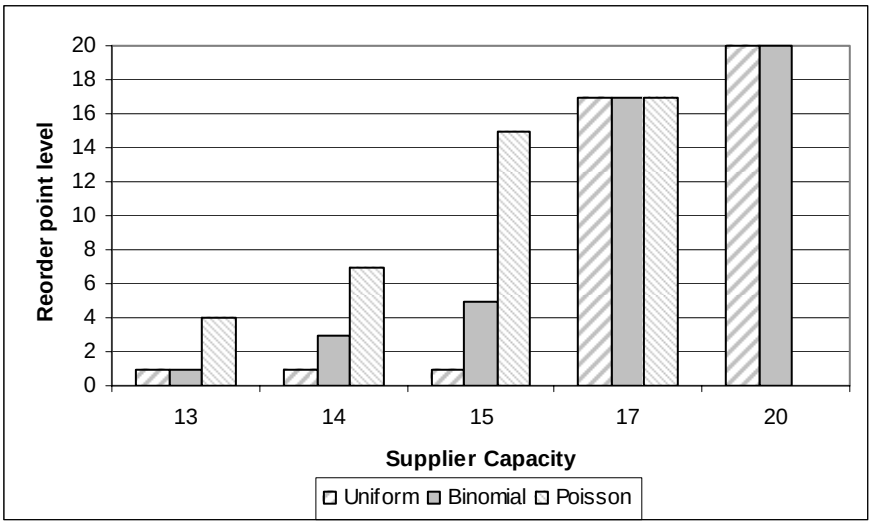


Figure 7. 7 Value of (*s*) As a Function of Supplier Capacity

When the retailer shares his demand and inventory position, the supplier is in effect making decisions after receiving information. He knows the retailer's demand, order quantity and his remaining inventory capacity once the retailer's order is filled. With this information, he is able to production in anticipation of the retailer's expected order quantity in the next period leading to a more balanced production schedule. Under this type of policy, his production resources are utilized 100% of the time. In effect, the supplier produces in advance of demand to mitigate the stock out as opposed to reacting after demand has depleted his inventory.

The structure of the optimal policy under information sharing also shows that as capacity becomes available, the supplier's myopic policy (used in the absence of information) and the completely observable policy become identical. The optimal policy found under information sharing policy is a state dependent modified echelon base stock policy. A modified base stock policy, as defined by Federgruen and Zipkin (1986), is a policy in which a base stock policy is followed when possible. When the prescribed production quantity exceeds the capacity, production is set to capacity. This is represented by the equation below were Z_j represents the echelon base stock quantity given the current retailer state is j and IP_e represents the current echelon inventory position at the beginning of the period.

$$z_s = \min\left(C_s, Z_j - IP_e\right) \qquad (7.1)$$

Figure 7.8 displays an example of a modified echelon base stock policy when the capacity of the supplier is 5 and capacity of the retailer is 8. In this example, the state dependent echelon base stock quantities Z_j are defined for each retailer state by the vector $\overline{Z} = (8,9,10,11,12,13,13,13)$. Figure 7.9 displays the information sharing policy for the case of binomially distributed demand over the recurrent state set. As the retailer's inventory level increases, the supplier starts to scale back his production quantity. This type of action is only achievable when information is shared because the production quantity is based on the inventory in the system, not just at the supplier.

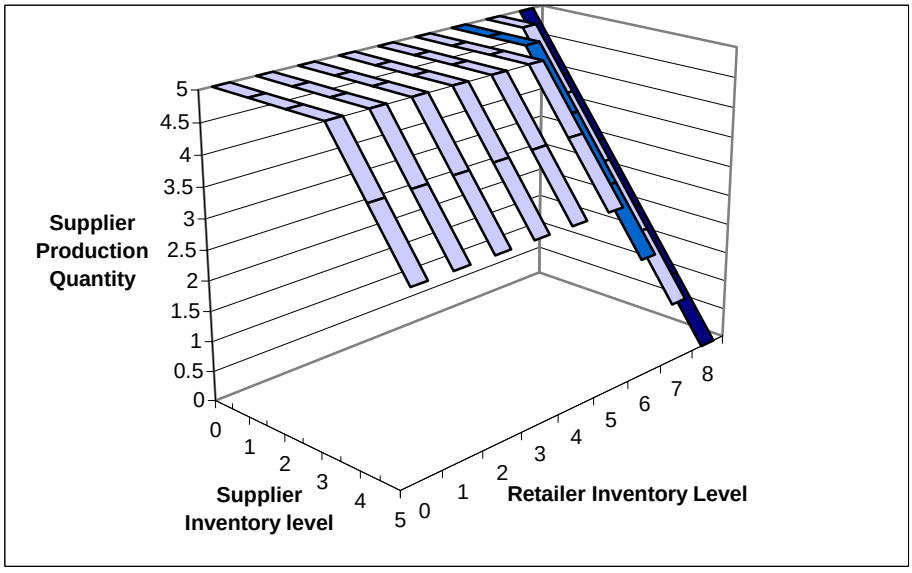


Figure 7. 8 Modified State Dependent Echelon Base Stock Policy

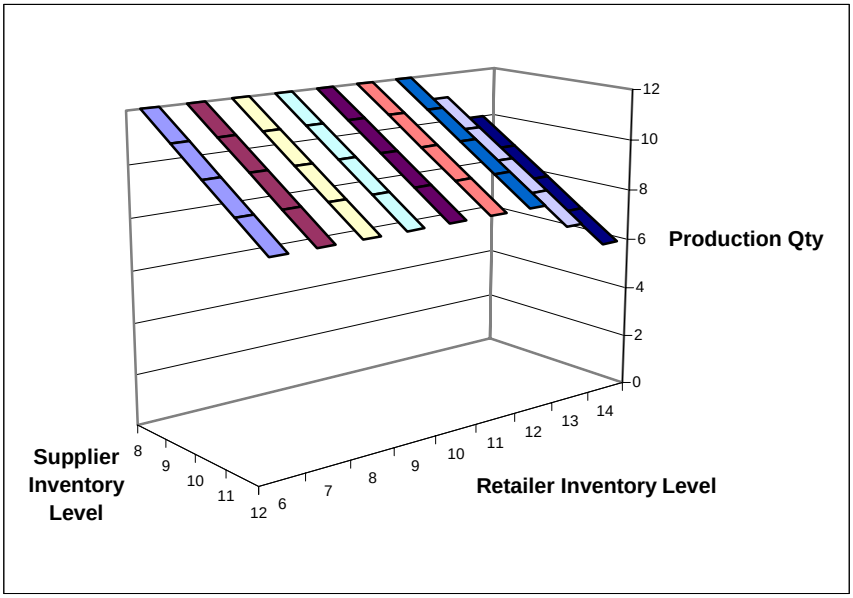


Figure 7. 9 Modified State Dependent Echelon Base Stock Policy for Recurrent States (Binomial demand distribution)

When supplier capacity is less than the mean demand, the supplier's ability to make a decision after receiving information allows production to occur in cases (states) where it otherwise wouldn't when information is not shared. When the supplier's capacity is equal to or greater than the retailer's capacity, the supplier can operate under a base stock policy without the use of information and satisfy all replenishment orders from the retailer. In this case, the production quantity for the supplier under the modified echelon base stock policy and the production quantity under the myopic base stock policy are nearly identical and there is little value in sharing information.

7.3.2.3 Sensitivity with respect to additional mean values

Figure 7.10 illustrates the structure of the VOI curve does not change as the mean demand changes. As the mean demand approaches the retailer's capacity limit, little value is obtained from sharing information. When the structure of the optimal policy is examined, the relative increase or decrease in the value of information can be attributed to two things; timing and scalability. First, the policies under no information sharing and information sharing are affected by the timing of the information. As stated earlier, when information is not shared, the available production is dependent upon the current inventory level of the supplier at the beginning of the period. When information is

shared, the available production is higher because the timing of the decision occurs after information is received. This enables production to occur in states that otherwise may not dictate it when information is not shared. Secondly, there is value in information when the supplier can scale back production based on the retailer's inventory level. When information is not shared, the supplier's production decision only changes as a function of the supplier's inventory level. Therefore, if multiple retailer states are recurrent when the supplier's inventory level is a particular value, the NIS production decision will be the same across all states while the IS production decision may vary. When the capacity exceeds the mean demand (as in the case when the mean is 12), the value in sharing information is high and largely attributed to the timing of decision as well as the ability to scale back production in certain retailer states. When capacity is tight relative to mean demand (as in the case when the mean is 18), the relative value of information sharing is lower. Since the demand rate is high, there is no opportunity to selectively scale back production in certain states and the supplier has to produce to capacity every time to mitigate the stock out cost. The little value that is achieved with sharing information is attributable only to the timing of the decision.

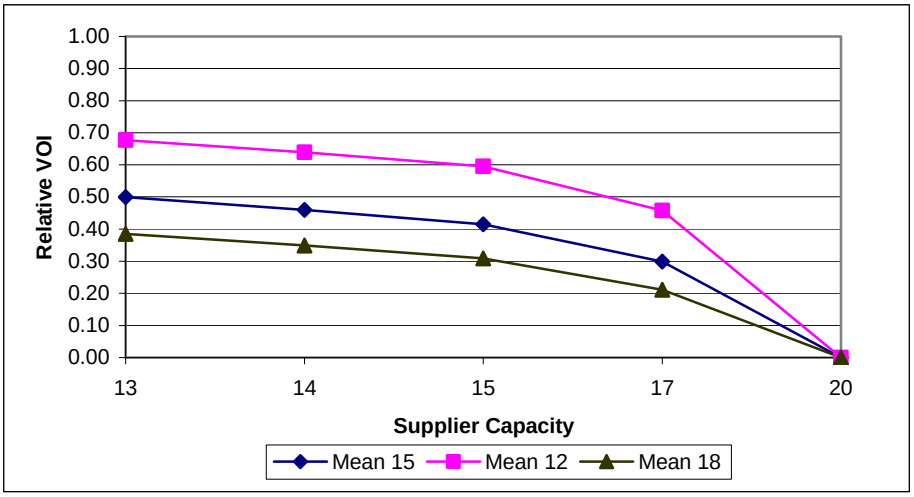


Figure 7. 10 Relative VOI for Binomial Distribution

7.4 Retailer policy effect on value of information sharing

7.4.1 Design of Experiment

In the previous experiments, the retailer implemented a base stock policy with base stock level equal to his maximum capacity. Therefore, the retailer is ordering with the supplier every period since the probability that zero units are demanded is very small. We now consider an (*s*,*S*) policy in which the retailer only orders when his inventory position falls below his safety stock level of *s*. An (*s*,*S*) policy is a base stock or order-up-to policy when *s*=*S*. Therefore, examining an (*s*,*S*) policy allows us to study the influence of (*S*-*s*) on the value of information sharing. When an (*s*,*S*) policy is used, the supplier may not receive demand information every period, depending upon the value of *s* with respect to the realized demand values. In this experiment, a value of *s* equal to $C_r/2$ is considered and all capacity and demand assumptions of section 7.3 hold.

7.4.2 Results

7.4.2.1 Value of information sharing

For the equivalent capacity and mean demand assumptions of section 7.3, the value of information sharing is significantly smaller when an (*s*,*S*) policy is used by the retailer. Figures 7.11 – 7.13 show the value of information sharing ranges from 0 to 50% when an order up to policy is used while the value is between 0 and 30% when an (*s*,*S*) policy is used. This reduction in VOI can be attributed to the loss of demand information being shared from a frequency standpoint. Several periods may elapse without ordering and therefore the supplier has less information from which to base his production decisions. He can either build up excess inventory and hold it or try to hold some intermediate amount of inventory that will mitigate the stock out cost. The latter choice is exactly what occurs when information is shared and is elaborated further in 7.4.2.2.

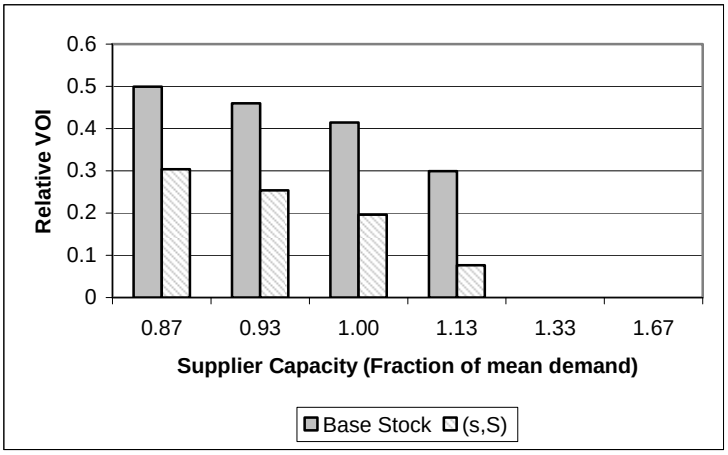


Figure 7. 11 Relative VOI for Binomial Demand

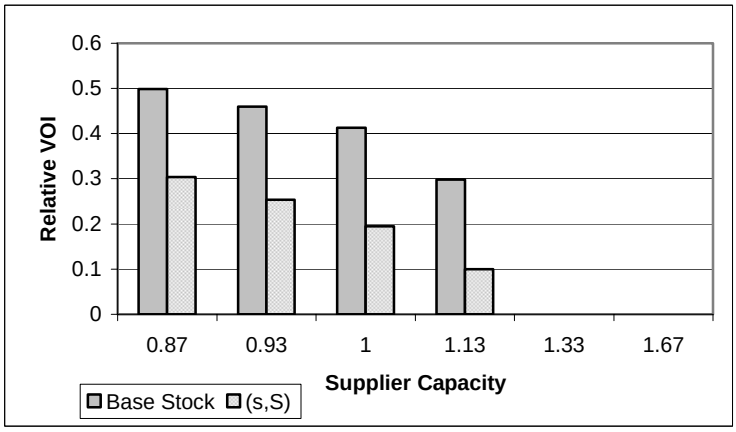


Figure 7. 12 Relative VOI for Uniform Demand

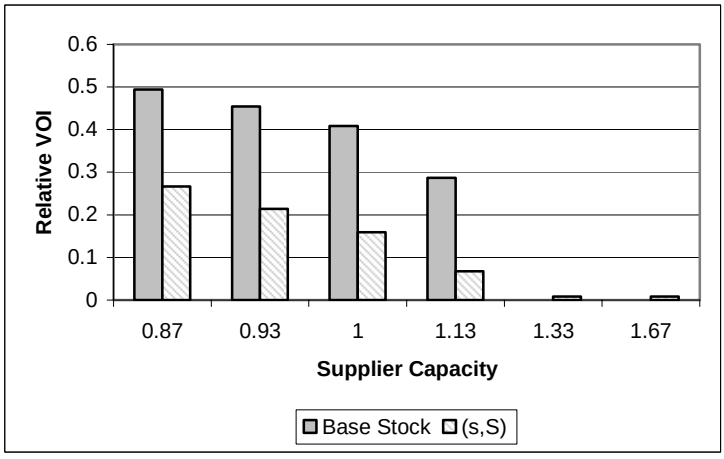


Figure 7. 13 Relative VOI for Poisson Demand

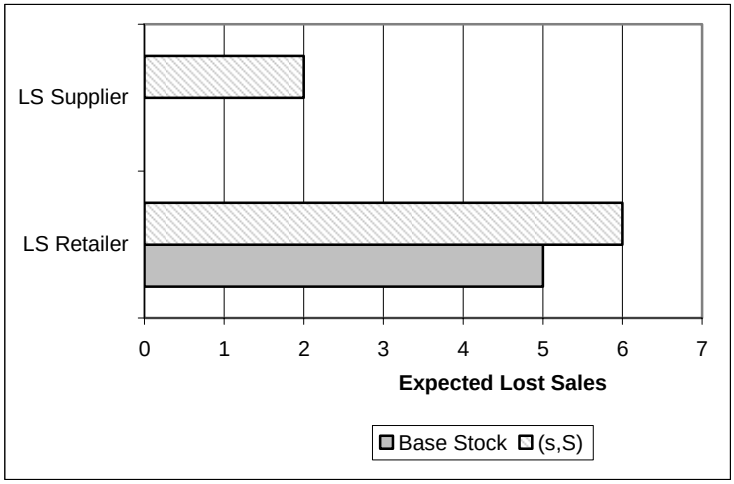


Figure 7. 14 Expected Lost sales with (*s*,*S*) and base stock

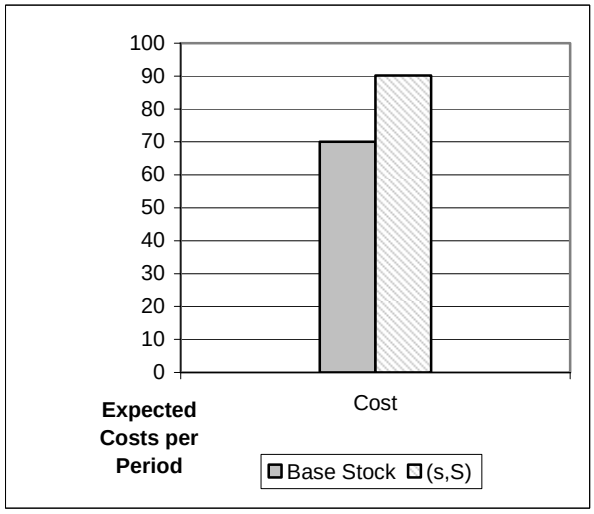


Figure 7. 15 Average Per Period Costs with (*s*,*S*) and Base Stock

The average per period costs and lost sales in the period are higher when an (*s*,*S*) policy is used as well. Figures 7.14 and 7.15 contrast the expected lost sales and expected per period supply chain cost between the two retailer inventory control policies when information is shared. The minimum lost sales for the supplier and the retailer as well as the lowest possible supply chain cost is achieved when the retailer uses an order-up-to policy.

Figure 7.16 shows expected production output when the demand is binomially distributed. The other demand distributions have the same form and are therefore not shown. As observed in the order-up-to case, the production output is higher when

information is shared. As more capacity becomes available, the supplier produces more and thus raises the echelon inventory level of the system, getting closer to the target value achieved with information sharing. This graph also shows that the output and associated information sharing policy is not changing. The decrease in relative VOI as capacity increases is due the decrease in cost associated with the changing structure of the no information sharing policy.

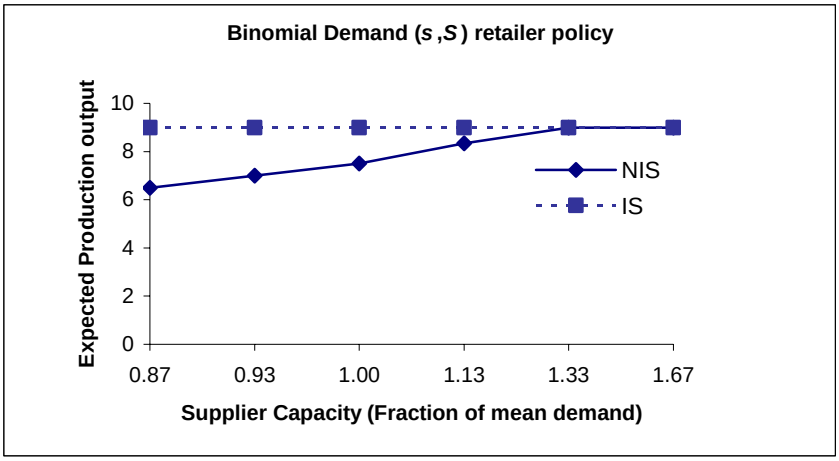


Figure 7. 16 Expected Supplier Production as a Function of Supplier Capacity

7.4.2.2 Optimal production control policy under steady state

One of the main differences in the relative VOI when different retailer policies are used is the manner in which the supplier responds once information from the retailer becomes available. In the case of a base stock policy, the supplier always uses the additional information to operate under a state dependent modified echelon base stock policy. Since the demand information is received every review period, he can respond every review period with the appropriate production quantity. In actuality, the supplier is ensuring that his pipeline plus on hand inventory always equals the retailer's base stock quantity, when capacity permits. However, when the retailer's inventory control policy is (*s*,*S*), a different operating policy under information sharing emerges. This policy illustrates how the supplier manages the inventory in the entire system to achieve the lowest possible cost.

As stated earlier, the optimal policy under no information sharing is an (*s*,*S*) policy. The policy parameters are nearly identical to those found under no information

sharing when the retailer's control policy is base stock for capacities 13, 14, 15 and 17. This is observed in all demand distributions of the experiment. Since the critical value of the policy parameter S is equal to the supplier's capacity C_s, the supplier fills replenishment orders that bring the retailer's inventory position above his reorder point. As a result, the retailer is not guaranteed to order the next period.

In contrast, the structure of the optimal policy under information sharing forces the retailer to order every period. The supplier's optimal control policy in steady state is to produce a fixed quantity every period. The quantity produced is insufficient to bring the retailer's inventory level above s if his demand during the period depletes his initial stock position. This guarantees that the retailer will order again in the next period. Figure 7.17, shows the retailer's steady state inventory position as a result of the supplier's production control policy. Once the retailer orders, his ending inventory position is based on what he receives from the supplier. Since the retailer is only giving him (S-s-1) units, the probability that the demand will exceed that amount is high (0.999). Therefore the retailer will always end the period with zero units and return to the same state at the beginning of the next period, from which he started in the previous period.

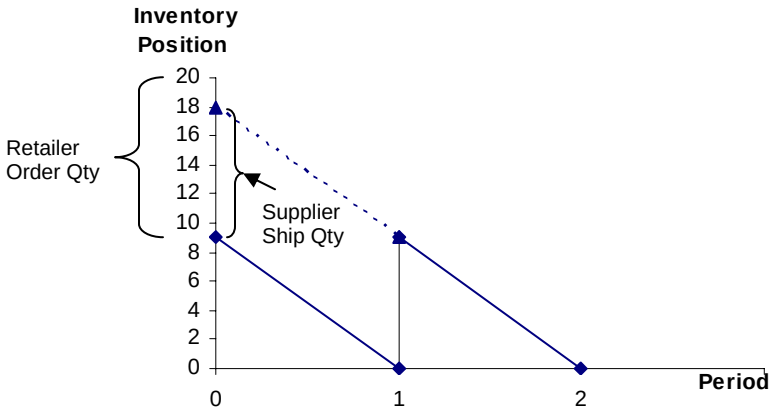


Figure 7. 17 Retailer Steady State Inventory Position each Review Period

The retailer order distribution resulting form this policy is shown in figure 7.18. This distribution shows the supplier's production control policy forces the retailer to order the same quantity every period, which triggers production for the supplier. Recall in section 7.3, when the retailer uses a base stock policy, the supplier satisfies all of the retailer's order quantity from inventory when information is shared. When an (*s*,*S*) policy is used by the retailer, controlling the retailer's shortage quantity results in lowering the expected per period costs of the system.

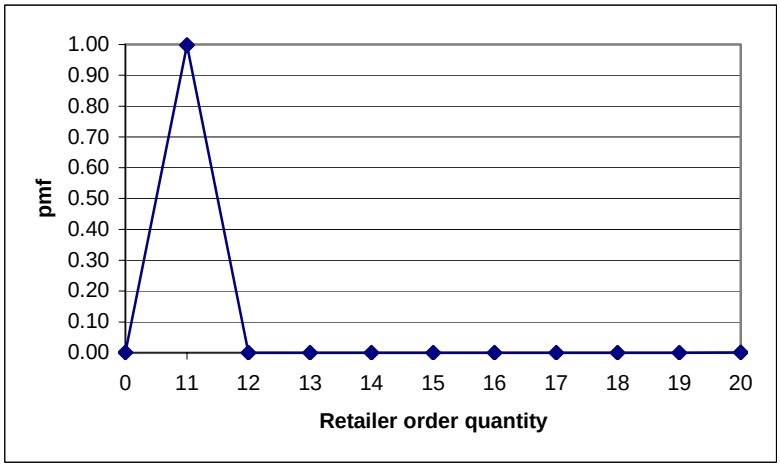


Figure 7. 18 Retailer Steady State Order Distribution

This information sharing policy structure is observed in all capacity instances except when the supplier and retailer capacity are equivalent. In this instance, additional

production quantities are considered reflecting the supplier's ability to produce in advance of demand. The states in which production occur correspond to states in which the retailer is not ordering. Therefore, the supplier is using the additional capacity to produce in advance of demand.

7.5 Information sharing effect on order variance

7.5.1 Design of experiment

In sections 7.2 and 7.3, experiments considered the effect capacity and the retailer's inventory control policy have on the value of information sharing. It is easy to determine the retailer's order distribution and associated moments under steady state optimal control to determine how information sharing effects the order variance. If the retailer operates under a base stock (C_r) policy, then his order quantity (z_r) given his current state is j is simply C_r-j. Therefore the probability of ordering C_r-j using the same retailer partitioning defined in section 7.1 is defined

$$\sum_{j \in R_k} x_j \left(C_r - j\right).$$

Similarly, if an (s,S) policy is used by the retailer, he orders 0 when j is greater than or equal to (S-s) and S-j otherwise.

7.5.2 Results

Figures 7.19 and 7.20 display the order distribution for one problem instance where the supplier capacity is 14 and the retailer capacity is 20. When a base stock policy is the used, the order mean and variance without information sharing are 13.4191 and 6.4312, respectively. When information sharing occurs, the order mean and variance are 10 and 0.75810, respectively.. From figure 7.19, the probability of receiving extreme order quantities (7 and 20) from the retailer are high when information is not shared and thus the order variance is very large. Sharing information reduces that order variance. Similar results are observed in figure 7.20 when an (s,S) policy is used. In this case, the extreme order quantities under no information share are 0 and 20. Again the order variance is reduced with information sharing from 96.641 to 0.188. Under similar demand and capacity assumptions, an (s,S) policy magnifies the variance of the order

distribution since the retailer is not ordering every period. Information sharing reduces that variance.

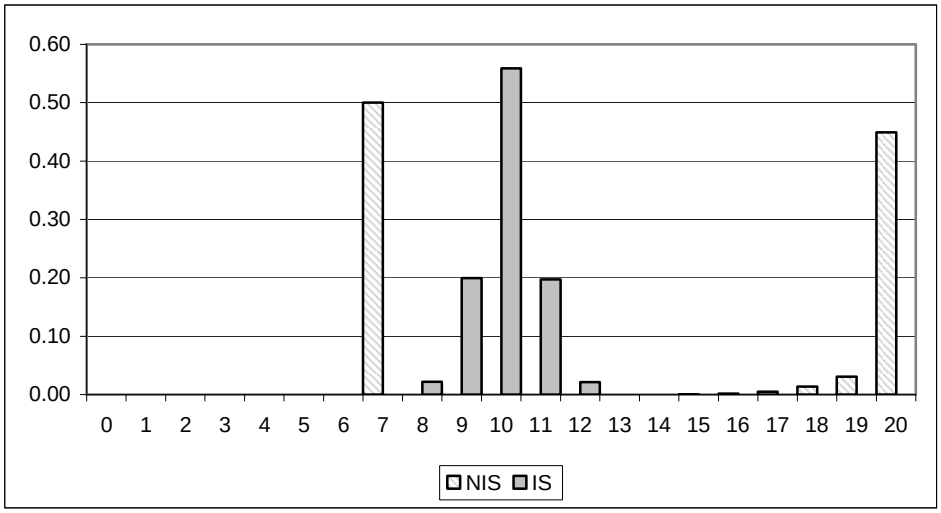


Figure 7. 19 Order distribution when demand is B(20,0.75) and retailer policy is Base Stock C_r

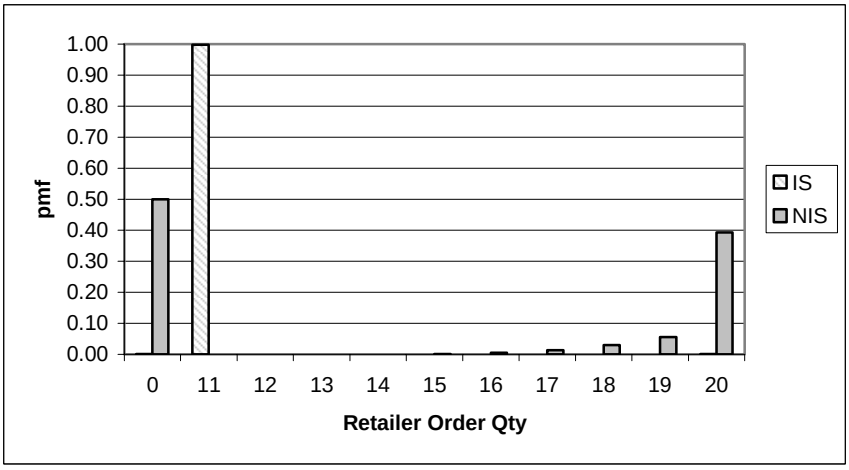


Figure 7. 20 Order distribution when demand is B(20,0.75) and retailer policy is (*s*,*S*)

7.6 Impact of costs on relative value of information sharing

7.6.1 Design of Experiment

The experiments in sections 7.2 through 7.4 did not consider any variable order/production costs or fixed cost of production. We considered costs that are increasing at the lower echelon. The unit holding and penalty costs at the supplier are cheaper than the costs at the retailer, which would be true in a production environment, where each stage adds value to the product. If variable order costs are added for the retailer, the relative value of information sharing will be affected. The addition of fixed production cost also will also affect the value of information. We are concerned with not only the affect on the value of information, but the change (if any) in the optimal policy structure for both information sharing and no information sharing.

7.6.2 Results

First we consider adding variable order costs to the retailer. The supplier's capacity is 13 and the retailer's capacity is 20. It was already shown earlier that the value of information sharing is high at lower capacities. The mean demand is equal to 15. As expected, increasing the variable order costs decreases the relative VOI, as shown in figure 7.21. The supplier's production control policy is the same as the zero variable cost case for the respective problems.

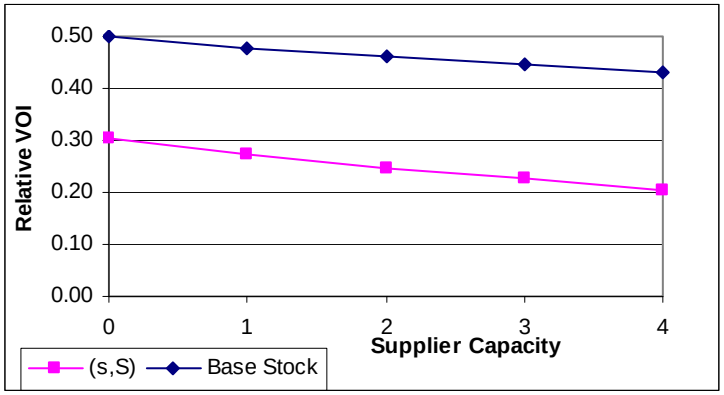


Figure 7. 21 Relative VOI as function of variable order cost, Supplier Capacity=13,

Similarly, when a fixed cost of production is added for the supplier, figure 7.22 shows the value of information decreases as the cost increases. The introduction of a fixed cost of production does not change the optimal policy under information sharing or no information sharing. The experiments were also executed for the case when the supplier capacity is 15 and similar results are obtained.

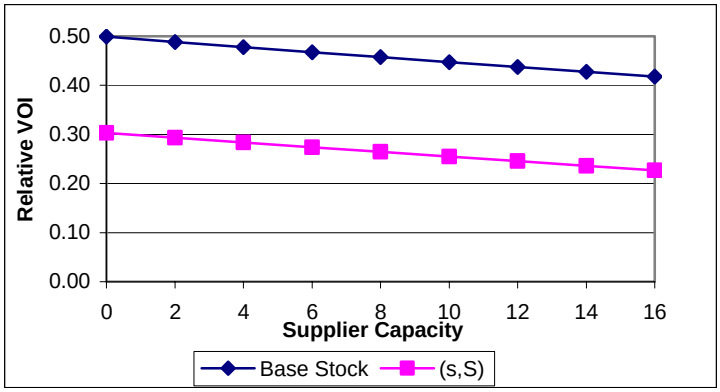


Figure 7. 22 Relative VOI as a Function of Fixed Production Cost, supplier capacity 13

7.7 Conclusions

The value of information sharing in a two stage supply chain with a capacitated supplier and retailer are affected by capacity, demand, cost structure and retailer's inventory control policy. In some instances the value of sharing information was as high as 50%. The relative value of information sharing is smaller when the retailer uses an (*s*,*S*) policy instead of a base stock policy. With the (*s*,*S*) policy, demand information is not passed every period and thus affects how the supplier can manage the inventory level and costs in the system. Overall, the retailer order variance is magnified and the expected per period costs are higher as a result.

Both the supplier and the retailer benefit from sharing information. The retailer is directly affected by not sharing information with the supplier in terms of their ability to satisfy their end item demand. The expected production of the supplier affects what the retailer is able to deliver which equates to lost sales. It is beneficial for the retailer to pass demand information every period. Whether or not this is information exchange

occurs through their inventory control policy or some other 'information' policy combined with their inventory control policy is an area of future work.

The amount of available capacity influences the value of information sharing. At low capacities, when information is not shared the supplier can't accommodate the demand from the retailer and much of the demand is lost thus impacting the retailer's responsiveness to his own demand requirements. Sharing information allows him to make the best use of his capacity. Increasing his capacity allows him to reach the maximum production output and system wide cost reduction. Once this maximum output is reached, any additional capacity results in no further cost reduction and the value of information sharing decreases. If the capacity constraint of the retailer is removed, increasing the supplier's capacity may result in further cost reduction and increased value of sharing information. This is an area of future work.

When the structure of the optimal policy is examined, the additional information received by the supplier allows him to produce in advance of future demand. This helps to mitigate the costs incurred from holding excess inventory and stocking out. As additional costs are added into the model, the relative VOI decreases. Although no change in the policy structure is observed under the assumptions of this model, other cost models and information sharing assumptions may induce changes and is an area of future work. It is interesting to see under what cost structure and assumptions production will be delayed several periods. Previous research on supply chain information sharing intimate sharing information is beneficial because the supplier can delay production. In the assumptions made in this study, the information sharing policy is one of constant production, even when fixed production costs are considered.

Modeling the problem as a completely observable MDP and restricted observation MDP is fairly simple and easy to interpret. The cost structure can be easily adapted to consider the retailer's cost only, supplier's cost only, or total supply chain costs. In addition, several performance measures can be calculated once the steady state distribution is known such as throughput, utilization, average inventory and lost sales to name a few. Further insight can be obtained by examining the optimal control policy of the supply chain member(s) being optimized. The experiments show that there is some

benefit in sharing information and that the information can allow the supplier to take a proactive approach to inventory management instead of reactive approach.

Chapter 8 Conclusions and Future Work

8.1 Conclusions

This research presents a new approach to measuring the value of information sharing in a supply chain. Using a completely observable Markov Decision model and a Markov Decision model with restricted observations is an ideal method for studying systems with and without perfect information. No assumption has to be made about the structure of the policy in order to find the value of information sharing. In this case, only the retailer's policy structure is assumed to be known but this constraint can be easily relaxed as well. This research shows that there is value having the retailer share information with the supplier. The magnitude of that value depends on the cost and capacity assumptions of the model. In addition, we describe that some periods of delayed production can be optimal instead of constant production.

8.2 Additional Research

The obvious area for additional research is to relax the assumption that the retailer policy is known. The retailer policy can be optimized as the supplier policy is being optimized. Whether or not the retailer needs backward information from the supplier or forward information from his customer can be considered.

In addition, this research only considers the case where demand is lost. Additional models including backlogging of demand and increased lead-time can be considered to understand the value of information sharing in those settings.

This research only touched on the structure of the optimal policy under information sharing and no information sharing. Further analysis can be done to determine the relationship between the critical values of the inventory control policy as a function of the retailer demand.

Additional models considering the timing of information are also important areas of work. This research showed how the retailers order policy and the value of (S-s) affect the value of information sharing. Additional policies that address timing of information flow if the retailer is using an (s,S) policy can be considered.

8.3 Stochastic Games

A Markov Decision Process is the simplest form of a stochastic game. There is additional research being done on information sharing from the game theoretic approach to address competing objectives between different supply chain members. The applicability of more complex models to this area can also be considered.

References

Cachon,G, and M.Fisher. 2000. Supply chain inventory management and the value of shared information. *Management Science* 46, no. 8: 1032 -1048.

Chen, F. 1998. Echelon Reorder Points, Installation Reorder Points, and the Value of Centralized Demand Information. *Management Science* 44, no. 12: S221-S234.

Clark, A., and H. Scarf. 1960. Optimal Policies for a Mulit-Echelon Inventory Problem. *Management Science* 6, no 4:475-490.

Dietz, H.M, and V. Nollau. 1983. *Markov Decision Problems with countable State spaces*. Akademi-Verlag, Berlin.

D'Amours, S, B. Montreuil, P. Lefrancois, and F. Soumis.1999. Networked Manufacturing: The impact of information sharing. *International Journal of Production Economics* 58: 63 – 79.

Ding, F., T. Hodgson, and R. King. 1988. A methodology for computation reduction for specially structured large scale Markov decision problems. *European Journal of Operational Research* 34: 105-112.

Federgruen, A., and P. Zipkin. 1986. An Inventory Model with Limited Production Capacity and Uncertain Demands I. The Average-Cost Criterion. *Mathematics of Operations Research* 11, no 2: 193-206.

Gallego, G., and O. Ozer. 2003. Optimal Replenishment Policies for Multiechelon Inventory Problems under Advance Demand Information. *Manufacturing and Service Operations Management* 5, no 2:157-175.

Gavirneni, S., R. Kapuscinski, and S. Tayur. 1999. Value of information in capacitated supply chains. *Management Science* 45, no. 1: 16 - 24.

Gavirneni, S. 2001. Benefits of cooperation in a production distribution environment. *European Journal of Operational Research* 130: 612 - 622.

_____. 2002. Information Flows in Capacitated Supply chains with fixed ordering costs. *Management Science* 48, no. 5: 644-651.

Hastings, N., and D. Sadjadi. 1979. Short Communication: Markov Programming with Policy Constraints. *European Journal of Operational Research* 3: 253-255.

Hodgson, T.J. and M. Zaldivar. 1975. Rapid convergence techniques for Markov Decision Processes. *Decision Sciences* 6: 14-24.

Hodgson, T.J and G. Koehler. 1979. Computation Techniques for Large Scale Undiscounted Markov Decision processes. *Naval Research Logistics Quarterly*, 26, no 4: 587-594.

Hordijk, A., and J. Loeve. 1994. Undiscounted Markov Decision Chains with Partial Information; An algorithm for computing a locally optimal periodic policy. *Mathematical Methods of Operations Research* 40: 163-181.

Horvath, L. 2001. Collaboration: the key to value creation in supply chain management. *Supply Chain Management: An International Journal* 6, no. 5: 205 - 207.

Howard, R. 1960. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

_____. 1971. *Dynamic Probabilistic Systems Volume 1. Markov Models* . Wiley & Sons, New York.

Huang, G., J. Lau, and K. Mak. 2003. The impacts of sharing production information on supply chain dynamics: a review of the literature. *International Journal of Production Research* 41, no. 7: 1483 – 1517.

Kemeny, J.G, and J.L. Snell. 1960. *Finite Markov Chains*. Van Nostrand Reinhold company, New Jersey.

King, F., T.J. Hodgson, and R. King. 1988. A methodology for computation reduction for specially structured large scale Markov decision problems. *European Journal of Operational Research* 34:105-112

Lee, H., and S. Whang. 2000. Information sharing in a supply chain. *International Journal of Technology Management* 20, nos. 3 / 4: 373 - 387.

Lee, H., K. So, and C. Tang. 2000. The value of information sharing in a two-level supply chain. *Management Science* 46, no. 5: 626 – 643.

Li, L. 2002. Information sharing in a supply chain with horizontal competition. *Management Science* 48, no. 9: 1196 – 1212.

Lumus, R. and R. Vokurka. 1999. Managing the demand chain through managing information flow: Capturing "Moments of information". *Production and Inventory Management Journal* – First Quarter: 16 - 20.

Odoni, A. 1969. On Finding the Maximal Gain for Markov Decision Processes. *Operations Research* 17, no. 5. 857-860.

Raghunathan, S. 2001. Information sharing in a supply chain: A note when demand is nonstationary. *Management Science* 47, no. 4: 605 – 610.

Sahin, F., and E. Robinson. 2002. Flow Coordination and Information Sharing in Supply Chains: Review, Implications, and Directions for Future Research. *Decision Sciences* 33, no.4: 1-32.

Serin, Y., and Z. Avsar. 1997. Markov Decision Processes with Restricted Observations: Finite Horizon Case. *Naval Research Logistics,* 44: 439-456.

Serin, Y., and V.G. Kulkarni. 1995. "Implementable Policies: Discounted Cost Case" in W.J. Stewart (Ed.), *Computations with Markov Chains*. Kluwer Academic Publishers, Dordrecht.

Simchi-Levi, D., and Y. Zhao. 2000. The value of information sharing in a two-stage supply chain with production capacity constraints: the infinite horizon case. Working paper, Northwestern University, Evanston, IL.

_____. 2002. The value of information sharing in a two-state supply chain with production capacity constraints: the infinite horizon case. *Manufacturing and Service Operations management* 4, no.1: 21 – 24.

Simmons, Donald M. 1975. *Nonlinear Programming for Operations Research.* Prentice Hall, New Jersey.

Smallwood, R., and E. Sondik. 1973. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Operations Research* 21: 1071-1088.

Smith, J.L. 1971. Markov Decisions on a partitioned state space. *IEEE transactions on systems, man and cybernetics* SMC-1, no. 1,: 55-60.

Sondik, E. 1978. The Optimal Control of Partially Observable Markov Processs over the Infinite Horizon: Discounted Costs. *Operations Research* 26, no. 2: 282-304.

Wei, J., and L. Krajewski. 2000. A model for comparing supply chain schedule integration approaches. *International Journal of production research* 38, no 9: 2099-2123.

White, D.J. 1963. Dynamic Programming, Markov Chains, and the method of successive approximations. *Journal of Mathematical Analysis and Applications* 6:373-376.

Wolfe, P., and G.B. Dantzig. 1962. Linear Programming in a Markov Chain. *Operations Research* 10:702-710

Yu, Z., H. Yan, and T.C. Cheng. 2001. Benefits of information sharing with supply chain partnerships. *Industrial Management and Data Systems* 101, no. 3: 114 – 119.

Zhao, X., J. Xie, and W. Zhang. 2002a. The impact of information sharing and ordering

co-ordination on supply chain performance. *Supply Chain Management: An International Journal* 7 no. 1: 24 – 40

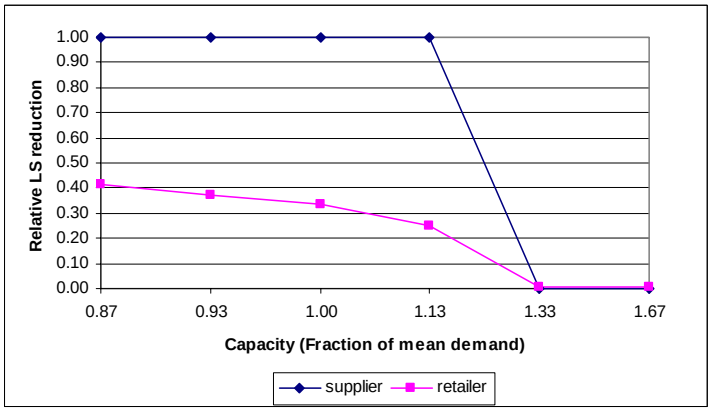
_____,J. Xie, and J.Leung. 2002b. The impact of forecasting model selection on the value of information sharing in a supply chain. *European Journal of Operational Research* 142: 321-344.

Appendix A Information Sharing Charts

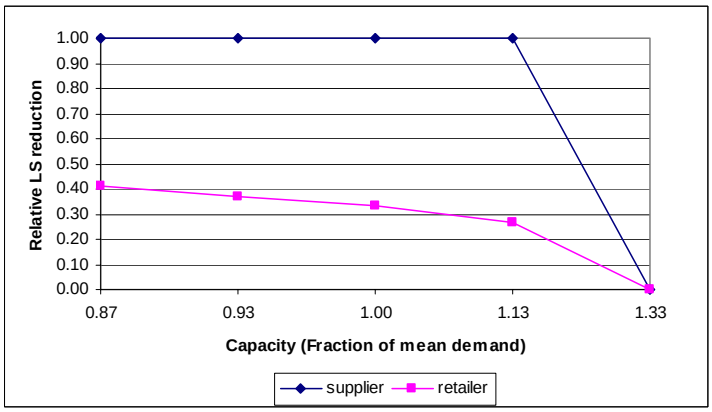
Information sharing charts when an Order-up-to policy is used by the retailer

The relative lost sales reduction is measured using the following equation.

$$\frac{\left(\overline{L}_{nis} - \overline{L}is\right)}{\overline{L}_{nis}}$$

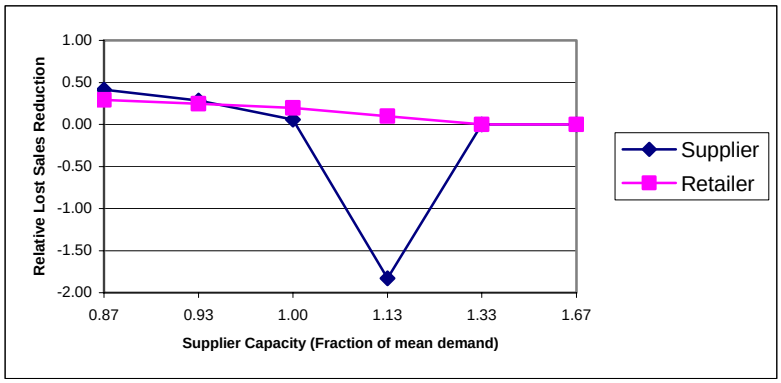


Relative Lost sales Reduction when demand is Uniform



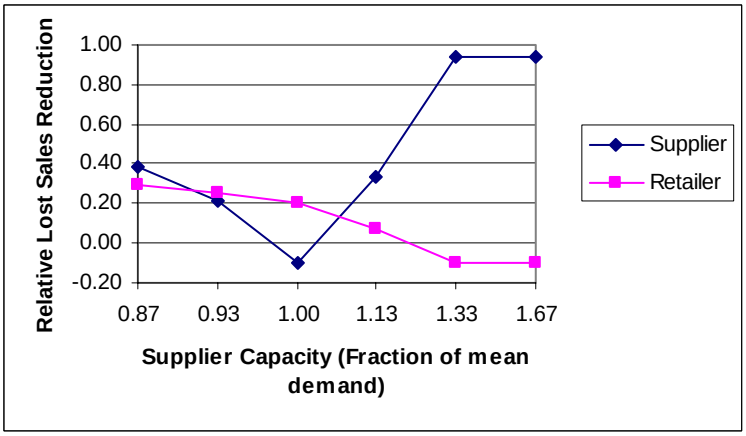
Relative Lost sales Reduction when demand is Poisson

Information sharing charts when an(s,S) policy is used by the retailer



-

Relative Lost sales Reduction when demand is Binomial



Relative Lost sales Reduction when demand is Uniform

Appendix B Glossary of Terms

α_{ka} : The probability of choosing action *a* for observation set *k*.

π : The implementable policy vector for the observed process with components $[\pi_1..\pi_K]$.

$\bar{x}$ Steady state information vector with components x_i denoting long run probability of being in state i.

A: The set of available actions for an instance of MDP/ROMDP {1…*M*}

A(*k*)*:* The set of admissible actions for observation set S_k. $A(k) \subseteq A$.

A_n*:* The action chosen at time *n*.

c_{ia} : The immediate reward associated with transitioning to state *i* under alternative $a \in A$. $c_{ia} = E\{C(X_n, A_n) \mid X_n = i, A_n = a\}$. In Howard's (1960) policy iteration algorithm, this quantity is denoted q_i^a .

C_s : Variable denoting the capacity of the supplier.

C_r : Variable denoting the capacity of the retailer.

g: The gain or long-run average cost associated with an implementable policy. Also referenced as $\Phi(\alpha)$ in non-linear programming formulation of ROMDP.

G(*i*): A function mapping a state *i* to a single observable output in the set *O*.

O: The set of observable outputs for an instance of ROMDP{1...*K*}.

$p_{ij}(a)$: The one step transition probability from state *i* to *j* under alternative $a \in A$.

$$p_{ij}(a) = P\{X_{n+1} = j \mid X_n = i, A_n = a\}$$

*pp*1 : Neighborhood construction scheme based on policy vector . Given vectors π and $\bar{\pi}$ and generated index *i* construct neighbor $\pi`$

$$\pi`_{[i]} = \bar{\pi}_{[i]}$$

$$\pi`_{[j]} = \bar{\pi}_{[j]} \quad \forall j \neq i$$

*pp*2 : Neighborhood construction scheme based on policy vector . Given vectors π and $\bar{\pi}$ and generated indices *i* and *j* construct neighbor $\pi`$

$$\pi`_{l} = \bar{\pi}_{[l]} \quad l \in \{i, j\}$$

$$\pi`_{[k]} = \bar{\pi}_{[k]} \quad \forall k \neq i, j$$

RDD: A randomized discrete distribution generated by creating random integers for a vector of length C_s+1 and then dividing each element by the vector sum. The resulting vector is a probability mass function for a distribution that takes on values between 0 and C_s.

S: The set of possible states for an instance of MDP/ROMDP{1…N}.

S_k: A given partition of the state space S satisfying $\{i:G(i) = k\}$. Also referred to as observation set.

X_n: A random variable denoting the state of the system at time n=0,1…

VOI_r : Relative value associated with sharing information calculated as

$$VOI_r = \frac{\phi\left(\alpha^*_{nis}\right) - \phi\left(\alpha^*_{is}\right)}{\phi\left(\alpha^*_{nis}\right)}.$$