Resource Allocation Problems in Stochastic Sequential Decision Making

by

Hariharan Lakshmanan

Bachelor of Technology, Civil Engineering
Indian Institute of Technology, Kharagpur (2001)
Master of Science, Massachusetts Institute of Technology (2004)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2009

Author ....
Department of Civil and Environmental Engineering
September 17, 2008

Certified by ....
Daniela Pucci de Farias
Lecturer of Mechanical Engineering
Thesis Supervisor

Certified by ....
David Simchi-Levi
Professor of Civil and Environmental Engineering and Engineering Systems
Thesis Supervisor

Accepted by ....
Daniele Veneziano
Chairman, Departmental Committee for Graduate Students

Resource Allocation Problems in Stochastic Sequential Decision Making

by

Hariharan Lakshmanan

Submitted to the Department of Civil and Environmental Engineering
on September 17, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in the field of Information Technology

Abstract

In this thesis, we study resource allocation problems that arise in the context of stochastic sequential decision making problems. The practical utility of optimal algorithms for these problems is limited due to their high computational and storage requirements. Also, an increasing number of applications require a decentralized solution. We develop techniques for approximately solving certain class of resource allocation problems that arise in the context of stochastic sequential decision making problems that are computationally efficient with a focus on decentralized algorithms where appropriate.

The first resource allocation problem that we study is a stochastic sequential decision making problem with multiple decision makers (agents) with two main features 1) Partial observability - Each agent may not have complete information regarding the system 2) Limited Communication - Each agent may not be able to communicate with all other agents at all times. We formulate a Markov Decision Process (MDP) for this problem. The features of partial observability and limited communication impose additional computational constraints on the exact solution of the MDP. We propose a scheme for approximating the optimal Q function and the optimal value function associated with this MDP as a linear combination of preselected basis functions. We show that the proposed approximation scheme leads to decentralization of the agents' decisions thereby enabling their implementation under limited communication. We propose a linear program, ALP, for selecting the parameters for combining the basis functions. We establish bounds relating the approximation error due to the choice of the parameters selected by the ALP with the best possible error given the choice of basis functions. Motivated by the need for a decentralized solution to the ALP, which is equivalent to a resource allocation problem with separable, concave objective function, we analyze a general class of resource allocation problems with separable concave objective functions. We propose a distributed algorithm for this class of problems when the objective function is differentiable and establish its convergence and convergence rate properties. We develop a smoothing scheme for non-differentiable objective functions and extend the algorithm for this case. Finally, we build on these results to extend the decentralized algorithm to accommodate non-negativity constraints on the resources. Numerical investigations on the performance of the developed algorithm show that our algorithm is competitive with its centralized counterpart.

The second resource allocation problem that we study is the problem of optimally accepting or rejecting arriving orders in a Make-To-Order (MTO) manufacturing firm. We model the production facility of the MTO manufacturing firm as a queue and view the time of the production facility as a resource that needs to be optimally allotted between current and future orders. We formulate the Order Acceptance Problem under two arrival processes - Poisson process (OAP-P), and Bernoulli Process (OAP-B) and formulate both problems as MDPs. We provide insights into the structure of the optimal order acceptance policy for OAP-B under the assumption of First Come First Served (FCFS) scheduling of accepted orders. We investigate a class of randomized order acceptance policies for OAP-B called static policies that are practically relevant due to their ease of implementation and develop a procedure for computing the policy gradient for any static policy. Using these results for OAP-B, we propose 4 heuristics for OAP-P. We numerically investigate the performance of the proposed heuristics and compare their performance with other heuristics reported in literature. One of our proposed heuristics, FCFS-ValueFunction outperforms other heuristics under a variety of conditions while also being easy to implement.

Thesis Supervisor: Daniela Pucci de Farias
Title: Lecturer of Mechanical Engineering

Thesis Supervisor: David Simchi-Levi
Title: Professor of Civil and Environmental Engineering and Engineering Systems

Acknowledgments

I would like to thank my advisor Prof. Daniela Pucci de Farias for her constant support for my research and for being a wonderful mentor. It has been a privilege working with her and learning from her. I am thankful to my advisor Prof. David Simchi-Levi for his support and guidance for my research. I have enjoyed the research discussions I had with him and learnt a lot from his insightful observations. I thank my defense committee member Prof. Steven Lerman for his valuable suggestions.

My special thanks to my roommate for 6 yrs M.N.Srinivas and office mate for 6 yrs, Sivaram. I will always remember and cherish the many wonderful moments we have had. My thanks to Deepak, J. Srini, Mani, Vikram, Vijay, Rags, Madhu for providing me such a wonderful atmosphere at MIT. My special thanks to Rags and Jaykumar for their insightful research discussions.

I thank my friends Senthil and Mythreyi for their wonderful company and support. I thank my friends Satish, Somu and Muthu who have been pillars of strength from my school days.

I thank my family for their unconditional support in all my endeavours.

Contents

List of Figures

Chapter 1

Introduction

Resource allocation is the art and science of allotting a limited amount of a resource among competing activities to optimize chosen criteria. Resource allocation problems arise in a wide variety of engineering and economic contexts. In this dissertation, we focus on resource allocation problems that arise in the context of stochastic sequential decision making. The computational effort required for optimal solution of these resource allocation problems is typically prohibitive. Also, an increasing number of applications require a decentralized solution. The focus of this dissertation is on providing efficient algorithms for solving certain class of resource allocation problems that arise in the context of stochastic sequential decision making problems, with an emphasis on decentralized algorithms where necessary.

1.1 Stochastic sequential decision making

Stochastic sequential decision making problems involve a system which evolves under a stochastic uncertainity as a result of the actions taken in stages by a decision maker. Markov Decision Processes (MDP) is a standard framework for studying stochastic sequential decision making problems. The state of the system of a MDP can be described through state variables and the set of all possible values that the state variables can take is called the state space. A decision maker typically has a choice of many actions for each state of the system. The goal of the decision maker is to find an optimal action for every state of the system for a chosen criteria.

Optimal algorithms are known for solving a wide variety of optimal control problems for MDPs. However, it is impossible to practically apply these algorithms due to their large computational and storage requirements. Hence it is important to develop computationally feasible heuristics for MDPs. We now briefly discuss the MDPs considered in this thesis, the associated resource allocation problems and the heuristics developed.

1.2 Resource allocation problem in MDPs involving multiple decision makers

For a large class of MDPs, there exists an optimal value function that associates a value with each state of the system that is a measure of the relative worth of that state. The optimal action to be taken in any state can be computed once the optimal value function is known. Approximate Dynamic Programming (ADP) refers to the collection of approximation techniques that aim to provide good, efficient solutions to MDPs. A popular idea in ADP is to approximate the optimal value function using approximation schemes with a few parameters and appropriately select the parameters to construct a good approximation. A linear programming based technique called approximate linear programming (ALP) falls under the category of ADP techniques that use value function approximation (see de Farias *et al.* [16]). ALP approximates the optimal value function as a linear combination of preselected basis functions, where each basis function associates a value to each state in the state space. It proposes a linear programming problem for selecting the weights for combining the basis functions. The first resource allocation problem studied in this thesis arises from an extension of ALP techniques to MDPs involving multiple decision makers. The decision makers would be called agents henceforth. A feature of these MDPs is that each agent may need to operate in an environment where communication with other agents is limited. It will be shown in the next chapter that in this case the linear programming problem for choosing the weights in ALP is equivalent to a resource allocation problem. Is is essential that this resource allocation problem is solved in a decentralized fashion. We propose and analyze a decentralized algorithm for a general class of resource allocation problems that includes the resource allocation equivalent of the ALP for MDPs with multiple agents.

1.3 Optimal Order Acceptance for a manufacturing firm

The second resource allocation problem studied in this thesis arises in the context of selective acceptance of orders to a manufacturing firm. We assume that the production capacity of the firm is fixed and hence the production time of the firm is a resource that must be judiciously allotted between various orders. A useful control available with the firm is to selectively accept the arriving orders. We formulate a MDP for optimally accepting arriving orders by modeling the production facility of the firm as a queue to which orders arrive via a known stochastic process. We study a related class of MDPs and derive computationally efficient heuristics for the order acceptance problem using solutions to the related problem.

1.4 Thesis organization and Contributions

We define the problem of optimal control of MDPs with multiple agents in Chapter 2. We discuss the computational difficulties in applying algorithms for optimal solution of this problem and establish the need for heuristics for obtaining good, efficient solutions. We describe an ALP based scheme for approximating important quantities related to the optimal solution of the MDP. We formulate a linear programming problem for obtaining the parameters related to the approximation scheme. We establish suitable error bounds relating the approximation from our ALP based scheme with the "best possible error" given the choice of the basis functions for the ALP. The problem setting requires a distributed solution to the linear programming problem for selecting the parameters of the approximation scheme. We establish the equivalence of this linear programming problem to a resource allocation problem with a separable concave objective function. This opens the possibility of using distributed algorithms developed for resource allocation problems for the linear programming problem of interest.

In Chapter 3, we define a general class of resource allocation problems with separable, concave objective functions, that includes the resource allocation equivalent of the linear programming problem for choosing the parameters of the approximation scheme. We propose and analyze a distributed algorithm for solution of this class of resource allocation problems by first assuming that the objective function is differentiable. We show convergence of our algorithm to an optimal solution and also derive convergence rate estimates. Based on this result, we develop smoothing schemes for non-differentiable objective functions and propose a distributed algorithm for solution of the resource allocation problem with separable, non-differentiable concave objective function. We also extend the results to a broader class of problems that include non-negativity constraints on the objective function.

In Chapter 4, we discuss the second resource allocation problem considered in this thesis, namely the optimal acceptance of orders for a manufacturing facility. We model the production facility of the firm as a queue and define the problem of optimally accepting arriving orders as a MDP. If the arrival of the orders to the firm is modeled by a Poisson process, the set of values that the variables representing the state of the system can take is not countable. We formulate a related MDP for the order acceptance problem that assumes a Bernoulli process for order arrival which is a problem with a finite state space.

In Chapter 5, we study the order acceptance problem with Bernoulli arrival process for orders. We propose an algorithm for finding the optimal solution to a special problem where there is no waiting room in the queue. We then establish some structural results of the optimal order acceptance policy assuming that the accepted orders are served on a first come first served basis. We define a class of order acceptance policies called the static policies which are easy to implement from a practical point of view. We establish a closed form expression for the expected average reward per time of a queue that processes only one type of order and use that expression to find the optimal policy among the class of static policies for a special problem.

In Chapter 6, we propose heuristics to the order acceptance problem with Poisson arrival of orders that makes use of the solution of an order acceptance problem with Bernoulli arrival process for orders. We evaluate the performance of the heuristics using numerical experiments and compare them to a heuristic proposed in the literature for the order acceptance problem with Poisson arrival process for orders.

We draw relevant conclusions and discuss possibilities for future research work in Chapter 7.

The results in Chapter 2 have previously appeared in [36] (Copyright (c) 2006 IEEE. Reused with permission. All rights reserved.). The results in Chapter 3 have previously appeared in [35] (Copyright (c)2008 Society for Industrial and Applied Mathematics. Reused with permission. All rights reserved.)

Chapter 2

Multi-Agent Sequential Decision Making Problems

2.1 Introduction

Markov Decision Processes (MDPs) offer a standard framework for studying stochastic sequential decision making problems. MDP has been the subject of extensive research over the last few decades and a large body of work covering various aspects of MDPs exist. A good introduction to the theory of MDPs is provided by [5], [42], [6]. The practical application of optimal algorithms for MDPs is limited due to the rapid growth in the computational requirements with the variables of interest, a problem known as the curse of dimensionality. In this work we are concerned with MDPs with multiple agents. To model a variety of practical situations, we assume that the communication between the agents may be limited, a factor that could affect their ability to coordinate in order to optimize decision making. We also assume that the agents may only have partial rather than complete information regarding the system at any time. These assumptions introduce additional computational difficulties in addition to the curse of dimensionality. In this chapter, we formulate the problem of multi-agent MDP's. We provide a linear programming based approximation framework for this problem. We show an error bound that relates the error induced by our approximation scheme with respect to the best possible error to be defined later in the chapter.

2.2 Problem description

We consider the problem of a network of agents operating in a stochastic environment, modeled as a MDP with the following characteristics. The MDP has a finite state space $\mathcal{S}$ and finite action space $\mathcal{A}^n$. An action $a = (a_1, \ldots, a_n)$ corresponds to a vector of individual actions taken by each of n agents. At each time stage, the system incurs a cost $g(x)$, where x denotes the state of the system. In the next time stage, the system transitions to state y with probability $P_a(x, y)$. We note that the system transition depends only on the current state and the joint action of the agents, a property known as the Markov property.

A *policy* u is a mapping from $\mathcal{S}$ to $\mathcal{A}^n$ specifying the action that each agent should take, conditioned on the current state of the system. We consider the problem of optimizing the expected sum of discounted rewards, over infinite horizon:

$$\min_{u \in \mathcal{U}} \mathrm{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) | x_0, x_{t+1} \sim P_{u(x_t)}(x_t, \cdot)\right], \tag{2.1}$$

simultaneously for all initial states x_0, where $\alpha \in (0, 1)$ is a *discount factor* that captures time preference and $\mathcal{U}$ is a set of admissible policies.

If $\mathcal{U}$ contains all possible mappings from states to actions, an optimal centralized control policy to be followed by the network of agents can be found via dynamic programming [5]. It can be shown that the problem of finding an optimal policy reduces to finding a solution to *Bellman's equation*

$$J(x) = \min_a \left\{ g(x) + \alpha \sum_y P_a(x, y) J(y) \right\}.$$

Bellman's equation has a unique solution denoted by J^*, which we refer to as the *optimal cost-to-*

go function. Another quantity of interest is the Q-function:

$$Q^*(x,a) = g(x) + \alpha \sum_y P_a(x,y) J^*(y).$$

An optimal policy u^* can be derived from Q^* according to

$$u^*(x) = \operatorname*{argmin}_a Q^*(x,a). \tag{2.2}$$

Stochastic dynamic programming (DP) [5] offers a systematic approach for finding the optimal centralized control policy to be followed by the network of agents. Nevertheless, application of DP presents two significant shortcomings. First, as mentioned before, dynamic programming is subject to the *curse of dimensionality*. In particular, computing and storing the optimal cost-to-go or Q-function requires an amount of resources that is at least linear on the cardinality of the state space. We expect that, for most problems involving networks of agents, the state of the system will typically include local states associated with each agent, and the cardinality of the state space will grow exponentially in the number of agents. Second, even if the cost-to-go or Q-function can be computed and stored efficiently, determining the optimal action at each time typically involves centralized operation. In particular, the optimal action is a function of the full state of the system, which is often not observable by any single agent and can only be determined through communication among all agents. In large networks, this may lead to prohibitive communication requirements. For instance, in computer networks involving a large number of servers ,or in teams of robots, or unmanned vehicles, the exchange of information that can occur within a certain period of time is limited by constraints of physical proximity and/or bandwidth of the channels available for communication.

While DP only generates centralized policies, finding an optimal decentralized policy directly is significantly harder. In particular, while DP has complexity that grows polynomially on the cardinality of the state space, finding an optimal decentralized policy is NP-hard [53]. Considering this, we propose a general framework that can be used to simultaneously address the curse of dimensionality and the need for decentralized control strategies.

As mentioned before, the agents are not assumed to have full information regarding the system at any time. We model this assumption regarding the partial information availability to the agents as follows. We assume that each agent i makes observations governed by a function $h_i : \mathcal{S} \mapsto \mathcal{O}$. In other words, when the state of the system is x, agent i observes $h_i(x)$. We model the communication between the agents at time t by an undirected graph $G(t)$ where nodes correspond to agents, and edges correspond to communication links. We assume that communication is symmetric, so that if agent i communicates with agent j, then agent j also communicates with agent i. We also assume that the union of the communication graphs is connected over any sufficiently large, bounded period of time. We formalize this assumption on the communication graphs in the next chapter.

We consider *approximation architectures* consisting of parametric classes of functions, which we use to approximate the Q-function. *Local approximation architectures* involve functions expressed as a sum of terms, each of which depends only on local information available to a given agent and its own action. Approximating the Q-function using a local approximation architecture is easily shown to give rise to decentralized control policies.

Given an approximation architecture, a function in that class must be selected as a suitable approximation to the Q-function. In this chapter, we propose and analyze a method for approximating the Q-function using a local, linear approximation architecture. Our method is based on the *linear programming approach to approximate DP*, or approximate linear programming (ALP) for short [16, 46]. We show that an error bound similar to that established for cost-to-go function approximation can be derived when linear programming is used to approximate the Q-function.

In many applications involving teams of agents, it is essential that the solution of the Q-function-fitting LP is obtained in a decentralized fashion. In the next chapter, we derive an interpretation of the Q-function fitting LP as a problem of resource allocation among the agents and propose decentralized algorithms for the solution of this problem. Before we describe our approximation approach we present a survey of related work.

2.3 Literature survey

A very good introduction to approximate DP techniques based on simulation of the underlying system can be found in [49] and [7]. The approximate DP methods described in these books are centralized. A good survey on centralized approximation techniques for MDPs can be found in [15]

The computation of optimal decentralized control policies is more complex than the computation of centralized control policies. Bernstein *et al.* [3] show that the computation of optimal control policies for a general finite horizon MDP with multiple agents where each agent has only an observation of the state is NEXP-complete. Goldman *et al.* [25] divide the problem of decentralized control of MDPs into various categories and provide complexity results for these categories.

Becker *et al.* [44] propose optimal algorithms for a class of multi-agent MDPs where there are local state variables corresponding to each agent and the evolution of these local state variables depends only on the action of the corresponding agent. ALP with local approximation architectures for decentralized control has been previously investigated in [13]. However, the approach presented requires solution of a centralized LP to generate an approximation to the Q-function. A hierarchical scheme for solving the ALP for *factored Markov decision processes* has also been proposed, but it requires that the agents form a network of fixed topology and take asymmetric roles [27]. Other approximate DP methods have been applied with empirical success to problems of routing and mobility control in ad-hoc networks (e.g., see [12]). However, these methods lack the convergence, error bounds or performance enjoyed by ALP methods.

2.4 Approximation architecture

We consider approximations to the Q-function given by linear combinations of *local basis functions* $\phi_{i,j} : \mathcal{O} \times \mathcal{A} \mapsto \Re$:

$$Q^*(x,a) \approx \sum_i \tilde{Q}_i(h_i(x), a_i, r_i) = \sum_{i=1}^{n} \sum_{j=1}^{K} \phi_{i,j}(h_i(x), a_i) r_{i,j}.$$

Note that $\phi_{i,j}$ depends on the global state x only through the information $h_i(x)$ available to agent i.

Recall that the optimal policy u^* is given by (2.2). Using $\tilde{Q}(\cdot,\cdot,r)$ as an approximation for Q^*, it is natural to consider following the policy

$$\begin{aligned} u(x) &= \operatorname{argmin}_a \{\sum_{i=1}^{n} \sum_{j=1}^{K} \phi_{i,j}(h_i(x), a_i) r_{i,j}\} \\ &= (\operatorname{argmin}_{a_i} \{\sum_{j=1}^{K} \phi_{i,j}(h_i(x), a_i) r_{i,j}\})_{i=1}^{n}. \end{aligned}$$

Hence each agent can make decisions $u_i(x)$ based only on the local information $h_i(x)$, without having to explicitly coordinate action choices with other agents. However, we note that the scheme still allows for coordination through appropriate choice of the state space and observation functions $h_i(x)$.

2.4.1 Approximate Linear Programming

Before introducing the linear programming method for fitting the Q-function, we define the operator H, which maps real functions on $\mathcal{O} \times \mathcal{A}$ to real functions on $\mathcal{O}$, as

$$(HQ_i)(h_i(x)) = \min_{a_i} Q_i(h_i(x), a_i).$$

The version of ALP we consider here requires that we deal with $H\tilde{Q}_i(\cdot, \cdot, r_i)$. Note that this function does not always admit a compact representation. In order to circumvent this issue, we consider approximations to $H\tilde{Q}_i$ given by

$$(H\tilde{Q}_i)(h_i(x)) \approx \tilde{J}_i(h_i(x), s_i) = \sum_{j=1}^{K} \psi_{i,j}(h_i(x))s_{i,j}.$$

Our scheme involves two forms of function approximation: on one hand, we propose approximating Q^* using a local, linear approximation architecture $\tilde{Q}_i(\cdot, \cdot, r_i)$. On the other hand, we also approximate $H\tilde{Q}_i$ using a linear approximation architecture $\tilde{J}_i(\cdot, s_i)$.

The choice of basis functions $\phi_{i,j}$ and $\psi_{i,j}$ involved in the definition of $\tilde{Q}_i$ and $\tilde{J}_i$ has central impact on the quality of the approximations that are generated. Appropriate choices of basis functions rely on problem-specific analysis and heuristics. Here, we assume that a set of basis functions $\phi_{i,j}$, $\psi_{i,j}$ have been specified in advance and focus on the task of choosing the weights $r_{i,j}$, $s_{i,j}$. Our analysis holds for an arbitrary choice of basis functions.

In order to find a set of weights $r_{i,j}$, $s_{i,j}$ leading to an appropriate approximation to the cost-to-go and Q-function, we consider the linear programming approach to approximate dynamic programming, here referred to as *approximate linear programming* (ALP). For the sake of simplicity, we introduce some matrix notation. We let $r_i \in \Re^K$, $s_i \in \Re^K$, $r \in \Re^{n \times K}$ and $s \in \Re^{n \times K}$ denote vectors $(r_{i,j})_{j=1,\ldots,K}$, $(s_{i,j})_{j=1,\ldots,K}$, $(r_{i,j})_{i=1,\ldots,n,j=1,\ldots,K}$, $(s_{i,j})_{i=1,\ldots,nj=1,\ldots,K}$, respectively. We also let $\Phi_i r_i$ and $\Psi_i s_i$ indicate functions $\sum_j \phi_{i,j}(h_i(x), a_i)r_{i,j}$ and $\sum_j \psi_{i,j}(h_i(x))s_{i,j}$. We consider the following linear programming (LP) problem for approximating the Q-function:

$$\begin{aligned}
\max_{r,s} \quad & \sum_{i,x,a} c(x,a)(\Phi_i r_i)(h_i(x),a) \\
\text{s.t.} \quad & g(x) + \alpha \sum_y P_a(x,y) \sum_i (\Psi_i s_i)(h_i(y)) \\
& \geq \sum_i (\Phi_i r_i)(h_i(x),a_i), \forall x, a \\
& (\Phi_i r_i)(h_i(x),a_i) \geq (\Psi_i s_i)(h_i(x)), \ \forall i, x, a_i.
\end{aligned} \tag{2.3}$$

The objective function coefficients $c(x,a)$ must be nonnegative and we assume without loss of generality that they add to one. The above LP has a number of variables that corresponds to the total number of basis functions $2nK$ used in the approximation scheme. The number of constraints grows linearly in the cardinality of the state and action spaces. However, because of the relatively small number of variables in the LP, we expect that most of the constraints will be redundant, and exact or approximate approaches can be derived for solving it efficiently. In particular, we expect that a constraint sampling scheme similar to that described in [17] for the case of cost-to-go function approximation would also apply to Q-function approximation. Alternative approaches that exploit problem-specific structure to deal with the constraints efficiently [28] could also be applied. In the next Section, we provide a bound on the error in the Q-function approximation generated by (2.3).

2.5 Error and Performance Analysis

An appealing feature of the ALP algorithm for cost-to-go function approximation is that it induces an error in the cost-to-go function approximation that is proportional in a certain sense to the smallest approximation error that can be achieved given the choice of the basis functions [16]. Specifically, if *some* linear combination of the basis functions is able to approximate the cost-to-go function well, then the error in the approximation provided by ALP cannot be much larger. In this section, we show that a similar result can be developed for the case of Q-function approximation.

In order to proceed with the analysis, we introduce some notation. For each policy u, we let $Q(x,u) = Q(x,u(x))$. We also let $P_u : \mathcal{S} \times \mathcal{S} \mapsto [0,1]$ denote the transition matrix associated with u, i.e., $P_u(x,y) = P_{u(x)}(x,y)$ for all x and y. For all $J : \mathcal{S} \mapsto \Re$ and $V : \mathcal{S} \mapsto \Re^+$, we define

$$\|J\|_{\infty,V} = \max_x \frac{|J(x)|}{V(x)},$$

with a similar definition for $\|\cdot\|_{V_i}$, $V_i : \mathcal{O} \mapsto \Re^+$. Moreover, for all $Q : \mathcal{S} \times \mathcal{A}^n \mapsto \Re$ and $V : \mathcal{S} \mapsto \Re^+$, we let

$$\|Q\|_{\infty,V} = \max_{x,a} \frac{|Q(x,a)|}{V(x)}.$$

We finally let

$$\|Q\|_{1,c} = \sum_{x,a} c(x,a)|Q(x,a)|.$$

The analysis is based on the use of *Lyapunov functions*, whose definition is given below. See [16] for a detailed discussion of their significance. For all $\mathcal{V} : \mathcal{S} \mapsto \Re^+$, let

$$\beta_V = \alpha \max_u \|P_u V\|_{\infty,V}.$$

Definition 2.5.1. *We call $V : \mathcal{S} \mapsto \Re^+$ a Lyapunov function if β_V is less than 1.*

We have the following result.

Theorem 2.5.1. *Suppose that there exist $V_i : \mathcal{O} \mapsto \Re^+$, $i = 1, \ldots, n$ such that:*

1. $V(\cdot) = \sum_i V_i(h_i(\cdot))$ *is a Lyapunov function;*

2. $(\Phi_i v_i^r)(\cdot, a_i) = (\Psi_i v_i^s)(\cdot) = V_i(\cdot)$ *for some v_i^r, v_i^s and all $a_i \in \mathcal{A}$.*

Let

$$\epsilon \;=\; \min_{r,s}\left(2\|Q^* - \sum_i \Phi_i r_i\|_{\infty,V} + \beta_V\left(\sum_i \|H\tilde{\Phi}_i r_i - \Psi_i s_i\|_{\infty,V} + \max_i \|H\tilde{\Phi}_i r_i - \Psi_i s_i\|_{\infty,V_i}\right)\right).$$

Also, for all i, let $V_i^Q = (\Phi_i v_i^r)$ and $V^Q = \sum_i V_i^Q$. Then the optimal solution $\tilde{r}$ of the approximate LP (2.3) satisfies

$$\|Q^* - \sum_i \Phi_i \tilde{r}_i\|_{1,c} \leq \frac{\|V^Q\|_{1,c}\epsilon}{1-\beta_V}.$$

Proof. Let $\hat{r}$, $\hat{s}$ achieve the minimum in the definition of ϵ. Let $\hat{J}_i = \Psi_i \hat{s}_i$, $\hat{Q}_i = \Phi_i \hat{r}_i$, $\hat{J} = \sum_i \hat{J}_i$ and $\hat{Q} = \sum_i \hat{Q}_i$. Let $\epsilon_1 = \max_u \|g + \alpha P_u \hat{J} - \hat{Q}(\cdot, u)\|_{\infty,V}$ and $\epsilon_2 = \max_i \|H\hat{Q}_i - \hat{J}_i\|_{\infty,V_i}$. Let $k^s = \frac{\epsilon_1+\epsilon_2}{1-\beta_V}$, $k^r = \frac{\epsilon_1+\beta_V\epsilon_2}{1-\beta_V}$. We first show that $\hat{r}_i - k^r v_i^r$, $\hat{s}_i - k^s v_i^s$ is a feasible solution for the approximate LP. For simplicity, let $\bar{Q}_i = \Phi_i(\hat{r}_i - k^r v_i^r) = \hat{Q}_i - k^r V_i^Q$, $\bar{J}_i = \Psi_i(\hat{s}_i - k^s v_i^s) = \hat{J}_i - k^s V_i$. For all u and $x \in \mathcal{S}$, we have

$$\begin{aligned}
g(x) + \alpha \sum_{y\in\mathcal{S}} P_{u(x)}(x,y)\bar{J}(y) \;&=\; g(x) + \alpha \sum_{y\in\mathcal{S}} P_{u(x)}(x,y)\left(\hat{J}(y) - k^s V(y)\right)\\
&\geq\; g(x) + \alpha \sum_{y\in\mathcal{S}} P_{u(x)}(x,y)\hat{J}(y) - \beta_V k^s V(x)\\
&=\; g(x) + \alpha \sum_{y\in\mathcal{S}} P_{u(x)}(x,y)\hat{J}(y) - \hat{Q}(x,u) + \hat{Q}(x,u)\\
&\qquad - k^r V(x) + (k^r - \beta_V k^s)V(x)\\
&\geq\; \left(-\epsilon_1 + k^r - \beta_V k^s\right) V(x) + \bar{Q}(x,u)\\
&=\; \left(-\epsilon_1 + \tfrac{\epsilon_1+\beta_V\epsilon_2-\beta_V(\epsilon_1+\epsilon_2)}{1-\beta_V}\right) V(x) + \bar{Q}(x,u)\\
&=\; \bar{Q}(x,u).
\end{aligned}$$

where the first inequality follows from the definition of a Lyapunov function. It follows that the first set of constraints in (2.3) are satisfied by $(\bar{J}_i, \bar{Q}_i)$. Moreover, we have

$$
\begin{aligned}
\bar{J}_i &= \hat{J}_i - k^s V_i \\
&= \hat{J}_i - H\hat{Q}_i + H\hat{Q}_i - k^r V_i + k^r V_i - k^s V_i \\
&= \hat{J}_i - H\hat{Q}_i + H(\hat{Q}_i - k^r V_i^Q) + k^r V_i - k^s V_i \\
&\leq (\epsilon_2 + k^r - k^s) V_i + H\bar{Q}_i \\
&= \left(\epsilon_2 + \tfrac{\epsilon_1 + \beta_V \epsilon_2 - \epsilon_1 - \epsilon_2}{1-\beta_V}\right) V_i + H\bar{Q}_i \\
&= H\bar{Q}_i.
\end{aligned}
$$

The third equality follows from the fact that $V_i^Q(h_i(x), a_i) = V_i(h_i(x))$ for all a_i. Hence the second set of constraints in (2.3) are also satisfied. We now note that the optimal solution $\tilde{r}$, $\tilde{s}$ of the approximate LP (2.3) minimizes $||Q^* - \sum_i \Phi_i r_i||_{1,c}$ over the feasible region [13]. Let $\tilde{Q} = \sum_i \Phi_i \tilde{r}_i$. It follows that

$$
\begin{aligned}
||Q^* - \tilde{Q}||_{1,c} &\leq ||Q^* - \bar{Q}||_{1,c} \\
&\leq ||Q^* - \hat{Q}||_{1,c} + k^r ||V^Q||_{1,c} \\
&\leq ||V^Q||_{1,c}(||Q^* - \hat{Q}||_{\infty,V} + k^r) \\
&= ||V^Q||_{1,c}\left(||Q^* - \hat{Q}||_{\infty,V} + \frac{\epsilon_1 + \beta_V \epsilon_2}{1 - \beta_V}\right)
\end{aligned}
\tag{2.4}
$$

The third inequality follows from the fact that for a given $x \in S$, $V^Q(x,a) = V(x)$ for all a.

Now note that

$$
\begin{aligned}
\epsilon_1 &= \max_u \|g + \alpha P_u \hat{J} - \hat{Q}(\cdot, u)\|_{\infty,V} \\
&\leq \max_u \|g + \alpha P_u \min_{u'} \hat{Q}(\cdot, u') - \hat{Q}(\cdot, u)\|_{\infty,V} + \\
&\quad + \max_u \|\alpha P_u \textstyle\sum_i (\hat{J}_i - H\hat{Q}_i)\|_{\infty,V} \\
&\leq \max_u \|\alpha P_u (\min_{u'} \hat{Q}(\cdot, u') - \min_{u'} Q^*(\cdot, u')) \\
&\quad + (Q^*(\cdot, u) - \hat{Q}(\cdot, u))\|_{\infty,V} + \beta_V \textstyle\sum_i \|\hat{J}_i - H\hat{Q}_i\|_{\infty,V} \\
&\leq (1 + \beta_V)\|Q^* - \hat{Q}\|_{\infty,V} + \beta_V \textstyle\sum_i \|\hat{J}_i - H\hat{Q}_i\|_{\infty,V}.
\end{aligned}
$$

The theorem follows from the above inequality and inequality (2.4). □

Theorem 2.5.1 implies that the approximation error $\|Q^* - \tilde{Q}\|_{1,c}$ provided by the approximate LP is proportional to the "best error" in approximating Q^* and J^* using the selected approximation architecture, and to the term $\sum_i \|H\Phi_i r_i - \Psi_i s_i\|_{\infty,V} + \max_i \|H\Phi_i r_i - \Psi_i s_i\|_{\infty,V_i}$. The bound is stated with respect to the objective function coefficients c and suggests that they can be used to emphasize approximation errors over different state-action pairs. Another important aspect of the result regards the role of Lyapunov functions. A straightforward way of ensuring existence of a suitable Lyapunov function is to let $\psi_{i,j}(\cdot) = \phi_{i,j}(\cdot,\cdot) = 1$ for some j and all i; it is easy to verify that the constant function is a Lyapunov function. However, we note that the Lyapunov function is used to weight errors over different state-action pairs in the constant ϵ. It can be shown that, in certain cases, a suitable Lyapunov function can be identified that captures structure of the system and ensures good scalability properties of the algorithm by de-emphasizing states that are less relevant for the decision-making process [16]. Moreover, another recent result shows that a different formulation of ALP can be used to relax the requirement for a Lyapunov function, while leading to similar bounds and scaling properties [18].

2.6 Equivalence of ALP to resource allocation problem

We have shown that the proposed approximation architecture using local basis functions has an appealing error bound. In problems involving multiple agents it is sometimes essential that the approximate LP (2.3) is solved in a decentralized fashion considering the communication constraints between the agents. In this section, we present a decentralized formulation for the approximate LP (2.3) that can be interpreted as a problem of resource allocation among the agents. We propose decentralized algorithms for a class of resource allocation problems in the next chapter which can be applied to the approximate LP (2.3).

Consider the following coupled optimization problems:

$$\max_{\lambda} \sum_{i=1}^{n} \mu_i(\lambda_i) \tag{2.5}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \lambda_i(x,a) \leq g(x), \forall x, a. \tag{2.6}$$

where

$$\mu_i(\lambda_i) = \max_{r_i, s_i} \sum_{x,a} c(x,a)(\Phi_i r_i)(h_i(x), a) \tag{2.7}$$

$$\text{s.t.} \quad \lambda_i(x,a) + \alpha \sum_{y} P_a(x,y)(\Psi_i s_i)(h_i(y)) \tag{2.8}$$

$$\geq (\Phi_i r_i)(h_i(x), a_i), \forall x, a$$

$$(\Phi_i r_i)(h_i(x), a_i) \geq (\Psi_i s_i)(h_i(x)), \forall x, a_i. \tag{2.9}$$

We have the following equivalence between problems (2.5) and (2.7) and the approximate LP (2.3).

Theorem 2.6.1. *1. $(\tilde{\lambda}, \tilde{r}, \tilde{s})$ is optimal for (2.5) and (2.7) only if $(\tilde{r}, \tilde{s})$ is optimal for the approximate LP (2.3).*

2. *$(\tilde{r}, \tilde{s})$ is optimal for (2.3) only if there exist a $\tilde{\lambda}$ such that $(\tilde{\lambda}, \tilde{r}, \tilde{s})$ is optimal for (2.5) and (2.7)*

proof of 1

Let $f(r, s)$ denote the value of the objective function for the approximate LP (2.3) for a given (r, s). Let $(\tilde{\lambda}, \tilde{r}, \tilde{s})$ be an optimal solution for (2.5) and (2.7). It is clear that $(\tilde{r}, \tilde{s})$ is a feasible solution for the approximate LP (2.3). Further, $f(\tilde{r}, \tilde{s}) = \sum_{i=1}^{n} \mu_i(\tilde{\lambda}_i)$. Consider any feasible solution $(\bar{r}, \bar{s})$ to the approximate LP (2.3). Define $\bar{\lambda}$ such that $\bar{\lambda}_i(x, a) = (\Phi_i \bar{r}_i)(h_i(x), a_i) - \alpha \sum_y P_a(x, y)(\Psi_i \bar{s}_i)(h_i(y))$. It is clear that $(\bar{\lambda}, \bar{r}, \bar{s})$ satisfies the constraints, (2.8), (2.9) and (2.6). Thus $f(\bar{r}, \bar{s}) \leq \sum_{i=1}^{n} \mu_i(\bar{\lambda}_i)$. Therefore $f(\tilde{\lambda}, \tilde{r}, \tilde{s}) = \sum_{i=1}^{n} \mu_i(\tilde{\lambda}_i) \geq \sum_{i=1}^{n} \mu_i(\bar{\lambda}_i) \geq f(\bar{r}, \bar{s})$. This establishes that $(\tilde{r}, \tilde{s})$ is an optimal solution to the approximate LP (2.3) proving the first part of the theorem.

proof of 2

Let $(\tilde{r}, \tilde{s})$ be an optimal solution to (2.3) and let $\tilde{\lambda}$ be defined such that $\tilde{\lambda}_i(x, a) = (\Phi_i \tilde{r}_i)(h_i(x), a_i) - \alpha \sum_y P_a(x, y)(\Psi_i \tilde{s}_i)(h_i(y))$. It is clear that $(\tilde{\lambda}, \tilde{r}, \tilde{s})$ satisfies the constraints, (2.8), (2.9) and (2.6). Consider an optimal solution $(\bar{\lambda}, \bar{r}, \bar{s})$ for (2.5) and (2.7). It is clear that $(\bar{r}, \bar{s})$ is a feasible solution for (2.3). Thus, $\sum_{i=1}^{n} \mu_i(\tilde{\lambda}_i) \geq f(\tilde{r}, \tilde{s}) \geq f(\bar{r}, \bar{s}) = \sum_{i=1}^{n} \mu_i(\bar{\lambda}_i)$ establishing the optimality of $(\tilde{\lambda}, \tilde{r}, \tilde{s})$ for (2.5) and (2.7).

Note that each LP (2.7) can be solved locally by the corresponding agent. We can think of $g(\cdot)$ as the total amount of resources to be assigned to each agent. Solving problem (2.5) corresponds to finding the optimal allocation, when the utility of resources to each agent is given by $\mu_i(\cdot)$. The following result gives more insight into the form of the utility functions $\mu_i(\cdot)$, establishing that (2.5) is a convex optimization problem. Another important consequence is that derivatives for (2.5) can be computed in a decentralized way.

Theorem 2.6.2. *1. For each i, μ_i is a concave, piecewise linear function. Its subgradients correspond to the optimal solutions of the dual of (2.7).*

2. *If $\psi_{i,\hat{j}}(\cdot) = \phi_{i,\hat{j}}(\cdot,\cdot) = 1$ for some $\hat{j}$, then for each value of λ_i the LP (2.7) has an optimal solution, hence its dual has a nonempty, bounded feasible region.*

Proof of 1: The result follows directly from LP duality theory (For example, Sections 5.2 and 5.3 of [9]) applied to (2.7). □

Proof of 2:

Consider a MDP where a cost of $\lambda_i(x, a)$ is incurred when the control action a is taken at state x. Consider approximating the optimal Q function and the optimal cost-to-go function for this problem using a linear approximation architecture using the basis functions ϕ_i, ψ_i and using the LP (2.7) for selecting the weights for combining these basis functions. Let the optimal Q function for this problem be denoted as $Q^*_{\lambda_i}$ to indicate its dependence of λ_i.

Define $\bar{r}_i$ such that $\bar{r}_{i\hat{j}} = 1$ and $\bar{r}_{ij} = 0$ for all other j. Similarly define $\bar{s}_i$ such that $\bar{s}_{i\hat{j}} = 1$ and $\bar{s}_{ij} = 0$ for all other j. Then $(\phi_i\bar{r}_i)(, a_i) = (\psi_i\bar{s}_i)(.) = 1$. Let $V_i = \psi_i\bar{s}_i$ and $V_i^Q = \phi_i\bar{r}_i$. It can be verified that V_i is a Lyapunov function. For some $\tilde{r}_i$, $\tilde{s}_i$, let $\tilde{J}_i = \psi_i\tilde{s}_i$ and $\tilde{Q}_i = \phi_i\tilde{r}_i$. Define $\epsilon_1 = \max_u \|g + \alpha P_u\tilde{J}_i - \tilde{Q}_i(\cdot, u)\|_{\infty,V_i}$ and $\epsilon_2 = \|H\tilde{Q}_i - \tilde{J}_i\|_{\infty,V_i}$. Let $k^s = \frac{\epsilon_1+\epsilon_2}{1-\beta_V}$, $k^r = \frac{\epsilon_1+\beta_V\epsilon_2}{1-\beta_V}$. Following the proof of Theorem (2.5.1), it can be shown that $\tilde{r}_i - k^r\bar{r}_i$, $\tilde{s}_i - k^s\bar{s}_i$ is a feasible solution for the LP (2.7). Also, it can be shown that for any feasible solution r_i, s_i for LP (2.7), $\phi_i r_i \le Q^*_{\lambda_i}$ (see proof of Theorem 1, Cogill *et al.* [13]) and hence $\sum_{x,a} c(x,a)(\phi_i r_i)(h_i(x), a) \le \sum_{x,a} c(x,a)Q^*_{\lambda_i}(x,a)$. Hence there exists an optimal solution to the LP (2.7). □

Considering problems (2.5)-(2.7) raises the possibility of applying decentralized resource allocation schemes to the approximate LP (2.3). This is the subject of the next chapter, where we propose decentralized algorithms for solution of a class of resource allocation problems that includes the resource allocation problem defined by (2.5) and (2.7).

Chapter 3

Decentralized algorithm for resource allocation problems with dynamic networks of agents

3.1 Introduction

We noted in Chapter 2 that for multi-agent MDPs, it is essential to solve the approximate LP (2.3) is a decentralized fashion. We showed that the approximate LP (2.3) is equivalent to a resource allocation problem with separable piecewise concave objective function. In this Chapter we study a general class of resource allocation problems that includes the resource allocation equivalent of the approximate LP. We first consider the case when the objective function is differentiable and propose a decentralized algorithm that converges to an optimal solution. We build on this result to provide a randomized decentralized algorithm that converges to an optimal solution of this class of resource allocation problems with probability one when the objective function is non-differentiable. We then extend these results to a broader class of resource allocation problems that includes non-negativity constraints on the resources.

3.2 Resource allocation problem formulation

In this section, we formulate a class of resource allocation problems. We consider the problem of n agents that share m common resources. Agent i has utility function f_i. The optimal allocation of resources for maximizing the average of the utilities among agents is given by the following optimization problem:

$$\begin{aligned} \max_{\lambda_i \in \Re^m, i=1,\ldots,n} \quad & f(\lambda) = \frac{1}{n}\sum_{i=1}^{n} f_i(\lambda_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \lambda_i = B, \end{aligned} \tag{3.1}$$

where $B \in \Re^m$ corresponds to the total amount of resources.

We propose decentralized, asynchronous algorithms for solution of (3.1). The first method applies in the case where $f_i, i = 1, \ldots, n$ are concave and differentiable, with Lipschitz continuous gradients. The second method applies in the case where $f_i, i = 1, \ldots, n$ are concave but not necessarily differentiable. We establish asymptotic convergence and convergence rates of both algorithms under mild conditions for communications among agents.

We showed that the approximate LP (2.3) is equivalent to the resource allocation problem defined by (2.5) and (2.7). We note that this resource allocation problem belongs to the class of problems defined by (3.1) with f_i being non-differentiable and hence the results established in the appropriate section of this chapter apply.

We recall, that at each iteration t, we model the communication between agents by an undirected graph $G(t)$ where nodes correspond to agents, and edges correspond to communication links. The decentralized algorithm for solving (3.1) in the case of differentiable utility functions has a simple gradient-ascent structure. Starting with an initial feasible resource allocation, agents trade resources with their neighbors at each iteration in proportion to the difference in gradient for the respective utility functions. The algorithm has a natural interpretation. The local gradient computed by each agent can be thought of as the price the agent is willing to pay for additional resources. At each iteration, agents trade resources with their neighbors in proportion to the prices each is willing to pay for the resources.

It can be shown that a large class of separable convex optimization problems with linear constraints can be transformed to equivalent resource allocation problems. The equivalence of the approximate LP to a resource allocation problem is an example. However the functions f_i in the transformed resource allocation problem are usually not differentiable. Motivated by this setting we consider the case where f_i is no longer differentiable, but has bounded subgradients. It is shown in this case that a randomized version of a decentralized subgradient-ascent algorithm converges with probability one to a near-optimal solution.

The subgradient-ascent algorithm for the case of non-differentiable utility functions can be interpreted as a stochastic approximation version of the gradient-ascent method for differentiable functions, applied to a smoothed version of the problem. The particular form of smoothing developed in this Chapter is motivated by several considerations. Adequate smoothing schemes must lead to a close approximation to the original function. Furthermore, as we build on the results for differentiable problems with Lipschitz continuous gradient, the gradient of the resulting smooth function must satisfy the same assumption with an adequate Lipschitz constant. Finally, another consideration in this Chapter is the computational effort involved in computing the gradient for the smoothed function. With this in mind, we propose a smooth approximation of the form $\hat{f}_i = \mathrm{E}[f_i(\lambda_i + Z_i)]$, where Z_i are vectors of zero-mean normal random variables. We show that, with an appropriate choice for the variance of Z_i, $\hat{f}_i$ is within ϵ of f_i and its gradient is Lipschitz continuous with a Lipschitz constant on the order of $O(\sqrt{\log m}/\epsilon)$, so that it scales gracefully on the dimension m of variable λ_i. In addition, this form of smoothing lends itself to application of a stochastic approximation scheme for gradient ascent which, at each iteration, only requires evaluation of a subgradient of f_i at a single point λ_i.

A comprehensive treatment of algorithms for various classes of resource allocation problems can be found in [51]. The algorithms introduced and analyzed in [51] are centralized in the sense that a central agent is assumed to have complete information about the problem and computes the optimal solution. In [1] and [29], decentralized resource allocation problems in the context of economics are investigated. The main difference in the approaches of [1, 29] as compared to the one presented here is the presence of a central agent who coordinates the computations performed by individual agents. A setting that is closer to that considered in this Chapter is presented in [47], which introduces a completely decentralized algorithm for a resource allocation problem with twice differentiable separable convex objective functions. The algorithm assumes a symmetric and fixed communication graph for the agents at all iterations and performs a gradient-projection at each iteration onto a subspace related to the communication graph. The same setting is considered in [38], which proposes a decentralized, weighted gradient algorithm for resource allocation problems with objective functions that are twice differentiable with bounded second derivatives. Dynamic communication graphs are considered in [32], which proposes an application-specific decentralized gradient algorithm for the problem of file allocation in distributed computer systems. Asynchronous gradient-descent methods are also considered in [52] for problems of unconstrained optimization with differentiable objective.

Most of the references regarding resource allocation problems in the literature, including the ones mentioned above, contain non-negativity constraints on the resources (i.e., they require $\lambda_i \geq 0, \forall i$), whereas in our formulation resources may be negative. In Section 3.5, we show how the results in this Chapter can be applied to problems with non-negativity constraints.

A distributed algorithm for non-differentiable optimization is presented in [39]. It is shown that a projected subgradient algorithm applied by each agent converges to the optimal solution. An important difference between the work presented in [39] and the work presented here is that the former requires that the long-run frequency of updates performed by each agent to be the same. Smoothing schemes for non-differentiable optimization can also be found in the literature. [40] proposes a smoothing scheme for functions f_i described as the maximum of differentiable functions. The smoothed function is within ϵ of f_i and has Lipschitz constant on the order of $O(1/\epsilon)$, independent of the dimensions of the problem. A caveat of this approach is that computing the gradient of the smoothed function may require multiple evaluations of subgradients of the original function. The particular form of smoothing considered here can also be found in the literature (see e.g. [50]); however, it does not contain results concerning the Lipschitz constant of the resulting smoothed function, which we develop in this Chapter.

3.3 Communication between agents

In this section we formalize our model of communication between agents. At iteration t, each agent i communicates with a set of agents denoted by $N_i(t)$. We assume that communication is symmetric, i.e. whenever agent i communicates with agent j, agent j also communicates with agent i. The communication between agents at time t can be represented by an undirected graph $G(t) = (N, E(t))$, where $N = \{1, \ldots, n\}$ represents the set of agents and the edge $(i, j) \in E(t)$ if and only if agent i communicates with agent j at time t. Let $E_{k,l} = \cup_{t=k}^{t=l-1} E(t)$. For a decentralized scheme to converge, the update of the variable associated with any agent must be periodically influenced by information from every other agent. This is ensured by the following assumption.

Assumption 3.3.1. *There exists a strictly increasing sequence $\{T_z\}$ of natural numbers with $T_1 = 1$ such that $G = (N, E_{T_z, T_{z+1}})$ is connected for all z and $(T_{z+1} - T_z) \leq \kappa$ where κ is some natural number.*

3.4 Decentralized Resource Allocation

We assume that (3.1) has an optimal solution. Let $\lambda \in \Re^{nm} = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ where $\lambda_i \in \Re^m$ for $i = 1, \ldots, n$.

Assumption 3.4.1. *There exists an optimal solution $\lambda^* = (\lambda_1^*, \lambda_2^*, \ldots, \lambda_n^*)$ to (3.1).*

For the rest of this Chapter, we let $||\cdot||$ denote the Euclidean norm.

3.4.1 The Differentiable Case

We now develop a decentralized algorithm for the case where f_i is concave and differentiable. We assume that the functions f_i have Lipschitz continuous gradients as formalized below.

Assumption 3.4.2. *There exists a constant $L > 0$ such that $||\nabla f_i(\lambda_i) - \nabla f_i(\bar{\lambda}_i)|| \leq L||\lambda_i - \bar{\lambda}_i||, \forall \lambda_i, \bar{\lambda}_i \in \Re^m$*

Recall that $f(\lambda) = \frac{1}{n}\sum_{i=1}^{n} f_i(\lambda_i)$. Hence

$$\begin{aligned} ||\nabla f(\lambda) - \nabla f(\bar{\lambda})|| &= \frac{1}{n}\sqrt{\sum_{i=1}^{n} ||\nabla f_i(\lambda_i) - \nabla f_i(\bar{\lambda}_i)||^2} \\ &\leq \frac{1}{n}\sqrt{\sum_{i=1}^{n} L^2||\lambda_i - \bar{\lambda}_i||^2} \\ &= \frac{L}{n}||\lambda - \bar{\lambda}|| \end{aligned}$$

The second equality follows from the fact that $||\lambda - \bar{\lambda}|| = \sqrt{\sum_{i=1}^{n} ||\lambda_i - \bar{\lambda}_i||^2}$. Hence $\frac{L}{n}$ is a Lipschitz constant for the function f. The decentralized algorithm that we develop is based on the following lemma, which characterizes an optimal solution to (3.1) when functions f_i are all differentiable.

Lemma 3.4.1. *A feasible solution λ^* of (3.1) is an optimal solution if and only if $\nabla f_i(\lambda_i^*) = \nabla f_j(\lambda_j^*)$ for all i, j.*

Proof. We first note that we can eliminate one of the variables in (3.1) to make it unconstrained. For instance, if we let $\lambda_n = B - \sum_{i=1}^{n-1} \lambda_i$, (3.1) is equivalent to

$$\min_{\lambda} \bar{f}(\lambda) = \frac{1}{n}\left(\sum_{i=1}^{n-1} f_i(\lambda_i) + f_n(B - \sum_{i=1}^{n-1} \lambda_i)\right)$$

This is an unconstrained concave and differentiable optimization problem, hence a solution λ^* is optimal if and only if $\nabla \bar{f}(\lambda^*) = 0$. Noting that

$$\nabla_{\lambda_i} \bar{f}(\lambda^*) = \frac{1}{n}(\nabla f_i(\lambda_i^*) - \nabla f_n(B - \sum_{j=1}^{n-1} \lambda_j^*)),$$

we conclude that λ^* is optimal if and only if

$$\nabla f_i(\lambda_i^*) = \nabla f_j(\lambda_j^*) = \nabla f_n(B - \sum_{j=1}^{n-1} \lambda_j^*) = \nabla f_n(\lambda_n^*), \forall i, j < n$$

□

Let λ_i^t be the value of the variable associated with agent i at iteration t. We consider the following gradient-ascent update rule for each agent i:

$$\lambda_i^{t+1} = \lambda_i^t + \gamma \sum_{j \in N_i(t)} \frac{1}{n}(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)) \tag{3.2}$$

Here γ is a common constant stepsize that all the agents use for updates. It should be noted that to perform updates at iteration t agent i uses only the gradient information corresponding to its neighbors $N_i(t)$ for iteration t. Each intermediate allocation λ^t generated by the algorithm is a feasible solution of (3.1):

Lemma 3.4.2. *Suppose λ^1 is a feasible solution for (3.1). Then λ^t, where λ_i^t is defined by (3.2), is a feasible solution to (3.1) for all t.*

Proof. Suppose λ^t is a feasible solution for (3.1). Then

$$\begin{aligned}
\sum_i \lambda_i^{t+1} &= \sum_i \lambda_i^t - \frac{\gamma}{n} \sum_i \sum_{j \in N_i(t)} (\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)) \\
&= B - \frac{\gamma}{n} \sum_{(i,j) \in E(t)} (\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t) + \\
&\qquad \nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)) \\
&= B
\end{aligned}$$

The second equality follows from the assumption that communication is symmetric. Thus λ^{t+1} is a feasible solution for (3.1) and the lemma follows by induction. □

In order to analyze the convergence properties of the proposed algorithm, it is convenient to define $\tilde{v}(\lambda)$ for any allocation λ as follows:

$$\tilde{v}_i(\lambda) = \sum_{j \in N} \frac{1}{n} (\nabla f_i(\lambda_i) - \nabla f_j(\lambda_j))$$

Note that $\tilde{v}(\lambda^t)$ is the direction of update when the communication graph $E(t)$ is complete. It can be verified that $\tilde{v}(\lambda^t)$ is also a scaled version of the projection of $\nabla f(\lambda^t)$ onto the subspace $\sum_{i=1}^n \lambda_i^t = B$, hence it represents the centralized update direction at time t. From Lemma 3.4.1 it can be seen that a feasible solution λ is optimal if and only if $||\tilde{v}(\lambda)|| = 0$. We now derive a theorem establishing convergence of the algorithm based on (3.2). Under mild conditions on the set of optimal solutions, convergence to optimality is guaranteed. We also derive an upper bound on the rate at which the sequence $\{||\tilde{v}(\lambda^{T_z})||\}$ converges to zero. Recall that T_z is a sequence of strictly increasing natural numbers such that the union of the communication graphs between iterations T_z and T_{z+1} is connected. In what follows, let $\tilde{v}^t = \tilde{v}(\lambda^t)$.

Theorem 3.4.1. *Suppose that Assumptions 3.4.1 and 3.4.2 hold. With a stepsize of* $\gamma = \frac{1}{2L}$,

1. *The sequence* $\{f(\lambda^t)\}$ *is monotonically non-decreasing.*

2. *The sequence* $\{\|\tilde{v}^{T_z}\|\}$ *converges to 0.*

3. $\min_{z=1,\ldots,p}(\|\tilde{v}^{T_z}\|^2) \leq \frac{3Ln^4\kappa(f(\lambda^*)-f(\lambda^1))}{4p}, \forall p.$

4. *If the set of optima is bounded, the sequence* $\{f(\lambda^t)\}$ *converges to* $f(\lambda^*)$.

The proof is based on a series of lemmas. Let the direction of update at time t be v^t. It can be seen from (3.2) that

$$v_i^t = \frac{1}{n} \sum_{j \in N_i(t)} (\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))$$

We first show that v^t is aligned to the direction of the gradient.

Lemma 3.4.3. $\nabla f(\lambda^t)^T v^t = \frac{1}{n^2} \sum_{(i,j)\in E(t)} \|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2$.

Proof. We have

$$\begin{aligned} \nabla f(\lambda^t)^T v^t &= \sum_{i \in N} \frac{1}{n} \nabla f_i(\lambda_i^t)^T \left(\frac{1}{n} \sum_{j \in N_i(t)} \nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t) \right) \\ &= \frac{1}{n^2} \sum_{i \in N} \nabla f_i(\lambda_i^t)^T \left(\sum_{j \in N_i(t)} \nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t) \right) \end{aligned} \tag{3.3}$$

Since communication is symmetric, for every term of the form $\nabla f_i(\lambda_i^t)^T(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))$ in the above summation, there is a corresponding term of the form $\nabla f_j(\lambda_j^t)^T(\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t))$. Hence,

$$\begin{aligned} \nabla f(\lambda^t)^T v^t &= \frac{1}{n^2} \sum_{(i,j)\in E(t)} \left(\nabla f_i(\lambda_i^t)^T \left(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\right) + \nabla f_j(\lambda_j^t)^T \left(\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\right)\right) \\ &= \frac{1}{n^2} \sum_{(i,j)\in E(t)} \|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2 \end{aligned}$$

□

We now prove a lemma that establishes a relationship between $||v^t||$ and $\nabla f(\lambda^t)^T v^t$. We can interpret $\gamma \nabla f(\lambda^t)^T v^t$ as the approximate increase in the objective of (3.1) when using the direction v^t and a sufficiently small step size γ.

Lemma 3.4.4. $||v^t||^2 \leq 2n\nabla f(\lambda^t)^T v^t$.

Proof. Using the Cauchy-Schwarz inequality, $(\sum_{i=1}^k c_i)^2 \leq k \sum_{i=1}^k c_i^2$,

$$
\begin{aligned}
||v_i^t||^2 &\leq |N_i(t)| \sum_{j \in N_i(t)} \frac{||\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)||^2}{n^2} \\
&\leq n \sum_{j \in N_i(t)} \frac{||\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)||^2}{n^2} \\
||v^t||^2 &= \sum_i ||v_i^t||^2 \\
&\leq n \sum_i \sum_{j \in N_i(t)} \frac{||\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)||^2}{n^2} \\
&= 2n \sum_{(i,j) \in E(t)} \frac{||\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)||^2}{n^2} \\
&= 2n \nabla f(\lambda^t)^T v^t
\end{aligned}
$$

The last equality comes from Lemma 3.4.3. □

We now prove a lemma that establishes a relationship between $||\tilde{v}^t||$ and $\nabla f(\lambda^t)^T \tilde{v}^t$

Lemma 3.4.5. $||\tilde{v}^t||^2 = n\nabla f(\lambda^t)^T \tilde{v}^t$.

Proof. We first have

$$
\begin{aligned}
\|\tilde{v}_i^t\|^2 &= \frac{\|\sum_{j\in N}(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))\|^2}{n^2} \\
&= \sum_{j\in N} \frac{\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2}{n^2} + 2 \sum_{((j,l)\in N^2, j<l)} \frac{(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))^T(\nabla f_i(\lambda_i^t) - \nabla f_l(\lambda_l^t))}{n^2} \\
&= \sum_{j\in N} \frac{\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2}{n^2} + \\
&\quad \sum_{((j,l)\in N^2, j<l)} \frac{\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2 + \|\nabla f_i(\lambda_i^t) - \nabla f_l(\lambda_l^t)\|^2 - \|\nabla f_j(\lambda_j^t) - \nabla f_l(\lambda_l^t)\|^2}{n^2} \\
&= (n-1)\left(\sum_{j\in N} \frac{\|(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))\|^2}{n^2}\right) - \sum_{((j,l)\in N^2, j<l, (j,l\neq i))} \frac{\|\nabla f_j(\lambda_j^t) - \nabla f_l(\lambda_l^t)\|^2}{n^2}
\end{aligned}
$$

$$
\begin{aligned}
\|\tilde{v}^t\|^2 &= \sum_{i\in N} \|\tilde{v}_i^t\|^2 \\
&= \sum_{i\in N} \left((n-1)\left(\sum_{(j\in N)} \frac{\|(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))\|^2}{n^2}\right) - \sum_{((j,l)\in N^2, j<l, j,l\neq i)} \frac{\|\nabla f_j(\lambda_j^t) - \nabla f_l(\lambda_l^t)\|^2}{n^2} \right)
\end{aligned}
$$

From the last equation, we note that $\|\tilde{v}^t\|^2 = \sum_{((i,j)\in N^2, i<j)} c_{ij} \left(\frac{\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2}{n^2}\right)$. To determine c_{ij}, note that the term $\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2$ appears with a coefficient $(n-1)$ in $\|\tilde{v}_i^t\|^2$ and $(n-1)$ in $\|\tilde{v}_j^t\|^2$ and with a coefficient -1 in $\|\tilde{v}_k^t\|^2$ for all $(k \in N, k \neq i, j)$. Hence, $c_{ij} = (n-1) + (n-1) - (n-2) = n$. Therefore,

$$
\begin{aligned}
\|\tilde{v}^t\|^2 &= \sum_{((i,j)\in N^2, i<j)} c_{ij} \left(\frac{\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2}{n^2}\right) \\
&= n \sum_{((i,j)\in N^2, i<j)} \frac{\|\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t)\|^2}{n^2} \\
&= n \nabla f(\lambda^t)^T \tilde{v}^t
\end{aligned}
$$

□

Consider a decentralized direction of update v^t derived from an arbitrary connected graph $G = (N, E(t))$. We now compare the ratio of the approximate increase in the objective of (3.1) using v^t as the direction of update and for a sufficiently small step size γ to the approximate increase in the objective using $\tilde{v}^t$ as the direction of update for the same step size. This ratio is given by:

$$\frac{(\nabla f(\lambda^t)^T v^t)}{(\nabla f(\lambda^t)^T \tilde{v}^t)} = \frac{\sum_{(i,j)\in E(t)}(\|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2)}{\sum_{((i,j)\in N^2, i<j)}(\|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2)} \tag{3.4}$$

The following lemma shows that this ratio is bounded away from 0 by a factor that only depends on the number of agents.

Lemma 3.4.6. *For all connected graphs $G = (N, E)$,*

$$\sum_{(i,j)\in E} \|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2 \geq \frac{8}{n^3} \sum_{((i,j)\in N^2, i<j)} \|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2$$

For any vector $\mathbf{x}$, let $(\mathbf{x})_k$ denote its k^{th} component. We recall that $\lambda_j^t \in \Re^m$ for $j \in N$ and note that $\frac{\sum_{(i,j)\in E}\|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2}{\sum_{((i,j)\in N^2, i<j)}\|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2}$ is of the form $\frac{\sum_{k=1}^m b_k}{\sum_{k=1}^m c_k}$, where $b_k = \sum_{(i,j)\in E}\left((\nabla f_j(\lambda_j^t))_k - (\nabla f_i(\lambda_i^t))_k\right)^2$ and $c_k = \sum_{((i,j)\in N^2, i<j)}\left((\nabla f_j(\lambda_j^t))_k - (\nabla f_i(\lambda_i^t))_k\right)^2$. Let $r_k = \frac{b_k}{c_k}$. We show that if $c_k > 0$, then $r_k \geq \frac{8}{n^3}$. We define $r(E) = \frac{\sum_{(i,j)\in E}(p_j - p_i)^2}{\sum_{((i,j)\in N^2, i<j)}(p_j - p_i)^2}$ for arbitrary values of the scalars p_i, $i = 1, \dots, n$, such that $\sum_{((i,j)\in N^2, i<j)}(p_j - p_i)^2 > 0$. We show that $r(E) \geq \frac{8}{n^3}$, which establishes that when $c_k > 0$, $r_k \geq \frac{8}{n^3}$. This result is based on a series of lemmas. We first establish that, for any fixed value of $p_i, i = 1, \dots, n$ the worst possible value of $r(E)$ is achieved when G corresponds to a chain whose nodes have monotone values of p_i. Then we compute the worst possible value of $r(E)$ with respect to possible values of p_i.

We also assume that $p_i \neq p_j$ for all $i \neq j$, without loss of generality; Since $\sum_{((i,j)\in N^2, i<j)}(p_j - p_i)^2 > 0$ by assumption, for any set of values p_i, $i = 1, \dots, n$, we can always perturb the values to make them strictly distinct while making $r(E)$ in the resulting graph arbitrarily close to that in the original problem.

Lemma 3.4.7. *The graph $G = (N, E)$ that minimizes r over all possible sets E, under the constraint that G is a connected graph, is a tree.*

Proof. Take an arbitrary graph (N, E), and suppose that it is not a tree. Then we can convert it into a tree (N, E') by removing some edges from E. It is clear that $r(E') \leq r(E)$, therefore (N, E) cannot be optimal. $\square$

Lemma 3.4.8. *If a certain graph (N, E) contains edges ij and jk such that $p_j < \min(p_i, p_k)$ or $p_j > \max(p_i, p_k)$, then it does not minimize r.*

Proof. Consider the first situation and suppose, without loss of generality, that $p_j < p_i < p_k$. Let $E' = E \backslash \{jk\} \cup \{ik\}$. The difference in the numerator of $r(E)$ and $r(E')$ is equal to $(p_j - p_k)^2 - (p_i - p_k)^2$ which is greater than 0. Therefore (N, E) cannot be optimal. Similar analysis holds when $p_j > \max(p_i, p_k)$. $\square$

Lemma 3.4.9. *If a node j contains more than two neighbors, then it has two neighbors i and k such that $p_j < min(p_i, p_k)$ or $p_j > max(p_i, p_k)$.*

Proof. Suppose that i, k, and l are neighbors of j. Then at least two among the three values p_i, p_k and p_l must be less than or greater than p_j. $\square$

Lemma 3.4.10. *Consider the chain that links nodes $1, \ldots, n$ in increasing order of p_i. Then it minimizes r over all possible connected graphs.*

Proof. From the previous lemmas, we conclude that the optimal graph is a tree. Moreover, each node in the optimal tree must have at most two neighbors. We conclude that the optimal graph is a chain. From Lemma 3.4.8, the nodes in the chain are in increasing or decreasing order of p_i, and the lemma follows. $\square$

Proof of lemma 3.4.6: Without loss of generality, suppose that $p_1 < p_2 < \cdots < p_n$. Let $\Delta_i =$

$p_{i+1} - p_i$. Note that, for all $j > i$, $p_j - p_i = \sum_{k=i}^{j-1} \Delta_k$. In view of the previous lemmas, we have, for every connected graph (N, E):

$$
\begin{aligned}
r(E) &= \frac{\sum_{(i,j)\in E}(p_i - p_j)^2}{\sum_{((i,j)\in N^2, i<j)}(p_i - p_j)^2} \\
&\geq \frac{\sum_{i=1}^{n-1}(p_{i+1} - p_i)^2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(p_i - p_j)^2} = \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\sum_{k=i}^{j-1}\Delta_k\right)^2} \\
&\geq \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=i}^{j-1}(j-i)\Delta_k^2} \\
&= \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{i=1}^{n-1}\sum_{k=i}^{n-1}\sum_{j=k+1}^{n}(j-i)\Delta_k^2} \\
&= \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{k=1}^{n-1}\Delta_k^2\sum_{i=1}^{k}\sum_{j=k+1}^{n}(j-i)} = \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{k=1}^{n-1}\Delta_k^2\sum_{i=1}^{k}\sum_{j=1}^{n-k}(j+k-i)} \\
&= \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{k=1}^{n-1}\Delta_k^2\sum_{i=1}^{k}\frac{(n-k)(n-k+1)}{2} + (k-i)(n-k)} \\
&= \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{k=1}^{n-1}\Delta_k^2\left(\frac{k(n-k)(n-k+1)}{2} + \frac{(n-k)(k)(k-1)}{2}\right)} \\
&= \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{k=1}^{n-1}\Delta_k^2\frac{k(n-k)(n)}{2}} \\
&\geq \frac{\sum_{i=1}^{n-1}\Delta_i^2}{\sum_{k=1}^{n-1}\Delta_k^2\frac{n^3}{8}} \\
&= \frac{8}{n^3}
\end{aligned}
$$

The second inequality follows from the Cauchy-Schwarz inequality.

We note from the definitions that if $c_k = 0$, then $b_k = 0$. Thus for $k = 1, \ldots, m$, either $b_k = c_k = 0$, or $r_k \geq \frac{8}{n^3}$. The Lemma is trivially true, if for $k = 1, \ldots, m$, $b_k = c_k = 0$. Suppose there exist some $\bar{k} \in (1, \ldots, m)$ such that $c_{\bar{k}} > 0$. Let K be the set of integers from 1 to m such that $c_k > 0$ for $k \in K$. K is not empty since it contains $\bar{k}$.

$$\begin{aligned}
\frac{\sum_{(i,j)\in E}(\|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2)}{\sum_{((i,j)\in N^2, i<j)}(\|\nabla f_j(\lambda_j^t) - \nabla f_i(\lambda_i^t)\|^2)} &= \frac{\sum_{k=1}^m b_k}{\sum_{k=1}^m c_k} \\
&= \frac{\sum_{k\in K} b_k}{\sum_{k\in K} c_k} \\
&\geq \frac{\sum_{k\in K} \frac{8}{n^3} c_k}{\sum_{k\in K} c_k} \\
&= \frac{8}{n^3}
\end{aligned}$$

□

Let E_{T_z} be a subset of the edge set $E_{T_z,T_{z+1}}$ such that the graph (N, E_{T_z}) is a tree. By assumption (3.3.1), the graph $(N, E_{T_z,T_{z+1}})$ is connected and so E_{T_z} is well defined. Let the decentralized direction of update derived using $G = (N, E_{T_z})$ at time T_z be denoted by $\bar{v}^{T_z}$. The following lemma shows that the approximate increase in the objective in period $[T_z, T_{z+1}]$ using the direction of update v^t and a sufficiently small step size γ is comparable to the approximate increase in objective when the direction $\bar{v}^{T_z}$ is used for update at time T_z.

Lemma 3.4.11. $\nabla f(\lambda^{T_z})^T \bar{v}^{T_z} \leq \frac{3}{2}\kappa \sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t$

Proof. We have

$$\begin{aligned}
\|\nabla f(\lambda^{t+1}) - \nabla f(\lambda^t)\|^2 &\leq \frac{L^2}{n^2}\|\gamma v^t\|^2 = \frac{1}{4n^2}\|v^t\|^2 \\
&\leq \frac{1}{4n^2} 2n(\nabla f(\lambda^t)^T v^t) = \frac{1}{2n}\nabla f(\lambda^t)^T v^t
\end{aligned} \tag{3.5}$$

The first inequality is true because of Assumption 3.4.2. The first equality is true because by assumption that the step size $\gamma = \frac{1}{2L}$. The second inequality follows from Lemma 3.4.4. Let $t_{T_z}^i$ be the earliest time between time periods T_z and $T_{z+1} - 1$ such that there is an edge $(i,j) \in E_{T_z}$ for agent i. It is clear that $T_z \leq t_{T_z}^i \leq T_{z+1} - 1$. Also, by definition, for $l = T_z, T_z + 1, \ldots, (t_{T_z}^i - 1)$, there is no edge $(i,p) \in E(l)$. Thus $\lambda_i^{t_{T_z}^i} = \lambda_i^{T_z}$ and $\nabla f_i(\lambda_i^{t_{T_z}^i}) = \nabla f_i(\lambda_i^{T_z})$. Letting $w_{ij}(t) =$

$\frac{1}{n}(\nabla f_i(\lambda_i^t) - \nabla f_j(\lambda_j^t))$, we have

$$
\begin{aligned}
\|w_{ij}(T_z)\| &= \frac{1}{n}\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{T_z})\| \\
&\leq \frac{1}{n}\left(\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{t^i_{T_z}})\| + \|\nabla f_j(\lambda_j^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{T_z})\|\right) \\
&\leq \frac{1}{n}\left(\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{t^i_{T_z}})\| + \sum_{t=T_z}^{t^i_{T_z}-1} \|\nabla f_j(\lambda_j^{t+1}) - \nabla f_j(\lambda_j^t)\|\right)
\end{aligned}
$$

From the Cauchy-Schwarz inequality,

$$
\begin{aligned}
\|w_{ij}(T_z)\|^2 &\leq \frac{(t^i_{T_z} - T_z + 1)}{n^2}\left(\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{t^i_{T_z}})\|^2 + \sum_{t=T_z}^{t^i_{T_z}-1} \|\nabla f_j(\lambda_j^{t+1}) - \nabla f_j(\lambda_j^t)\|^2\right) \\
&\leq \kappa\left(\frac{\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{t^i_{T_z}})\|^2}{n^2} + \sum_{t=T_z}^{t^i_{T_z}-1} \frac{\|\nabla f_j(\lambda_j^{t+1}) - \nabla f_j(\lambda_j^t)\|^2}{n^2}\right) \\
&\leq \kappa\left(\frac{\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{t^i_{T_z}})\|^2}{n^2} + \frac{1}{2n}\sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t\right)
\end{aligned}
$$

The last inequality comes from (3.5) and from the fact that $\|\nabla f(\lambda^{t+1}) - \nabla f(\lambda^t)\|^2 = \sum_{i=1}^n \frac{\|\nabla f(\lambda_i^{t+1}) - \nabla f(\lambda_i^t)\|^2}{n^2}$. We finally have

$$
\begin{aligned}
\nabla f(\lambda^{T_z})^T \bar{v}^{T_z} &= \sum_{(i,j)\in E_{T_z}} \|w_{ij}(T_z)\|^2 \\
&\leq \sum_{(i,j)\in E_{T_z}} \kappa\left(\frac{\|\nabla f_i(\lambda_i^{t^i_{T_z}}) - \nabla f_j(\lambda_j^{t^i_{T_z}})\|^2}{n^2} + \frac{1}{2n}\sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t\right) \\
&\leq \kappa\left(\sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t + \frac{n-1}{2n}\sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t\right) \\
&\leq \frac{3}{2}\kappa\left(\sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t\right)
\end{aligned}
$$

It is clear that Lemma 3.4.3 is valid for all decentralized directions of update v derived using some communication graph G where $v_i = \sum_{j \in N(i)} \frac{1}{n}(\nabla f_i(\lambda_i) - \nabla f_j(\lambda_j))$ and $N(i)$ is the set of neighbors of i in G. Hence Lemma 3.4.3 is valid for $\bar{v}^{T_z}$. The equality comes from lemma 3.4.3 with v^t replaced by $\bar{v}^{T_z}$. The second inequality comes from the fact that E_{T_z} is a subset of $E_{T_z,T_{z+1}}$ and from Lemma 3.4.3. The second inequality also depends on the fact that there are exactly $n-1$ edges in the set E_{T_z} as $G = (N, E_{T_z})$ is a tree. $\square$

Proof of Theorem 3.4.1:

Proof of 1: First note that

$$
\begin{aligned}
f(\lambda^{t+1}) - f(\lambda^t) &\geq \gamma \nabla f(\lambda^t)^T v^t - \frac{L}{2n}\|\gamma v^t\|^2 \\
&\geq \gamma \nabla f(\lambda^t)^T v^t - \frac{\gamma^2 L}{2n} 2n \nabla f(\lambda^t)^T v^t \\
&= \frac{1}{2L}\nabla f(\lambda^t)^T v^t - \frac{1}{4L}\nabla f(\lambda^t)^T v^t = \frac{1}{4L}\nabla f(\lambda^t)^T v^t
\end{aligned}
\tag{3.6}
$$

The first inequality comes from the descent Lemma for differentiable functions [4]. The second inequality comes from Lemma 3.4.4. The first equality comes from the fact that $\gamma = \frac{1}{2L}$. Since $\nabla f(\lambda^t)^T v^t$ is non-negative, the sequence $\{f(\lambda^t)\}$ is monotonic and non-decreasing establishing the first part of the theorem.

Proof of 2: Since (3.1) is assumed to have an optimal solution, $f(\lambda^t)$ is bounded from above. We conclude from the first claim that $\{f(\lambda^t)\}$ converges and $\{\nabla f(\lambda^t)^T v^t\}$ must converge to zero.

We now have

$$
\begin{aligned}
\|\tilde{v}^{T_z}\|^2 &= n \nabla f(\lambda^{T_z})^T \tilde{v}^{T_z} \\
&\leq \frac{n^4}{8}\nabla f(\lambda^{T_z})^T \tilde{v}^{T_z} \leq \frac{3n^4}{16}\kappa \left(\sum_{t=T_z}^{T_{z+1}-1} \nabla f(\lambda^t)^T v^t \right),
\end{aligned}
$$

where the first equality follows from Lemma 3.4.5, the first inequality follows from Lemma 3.4.6 and Lemma 3.4.3 and the second inequality follows from Lemma 3.4.11. The last inequality and the convergence of $\{\nabla f(\lambda^t)^T v^t\}$ to zero establishes the second part of the theorem.

Proof of 3: Note that

$$\begin{aligned}
f(\lambda^{T_{z+1}}) - f(\lambda^{T_z}) &\geq \frac{1}{4L}\sum_{t=T_z}^{t=T_{z+1}-1} \nabla f(\lambda^t)^T v^t \\
&\geq \frac{1}{6L\kappa}\nabla f(\lambda^{T_z})^T \bar{v}^{T_z} \\
&\geq \frac{4}{3Ln^3\kappa}\nabla f(\lambda^{T_z})^T \tilde{v}^{T_z} \\
&= \frac{4}{3Ln^4\kappa}\|\tilde{v}^{T_z}\|^2
\end{aligned}$$

The first inequality comes from (3.6). The second inequality comes from Lemma 3.4.11. The third inequality comes from Lemma 3.4.6 and Lemma 3.4.3 and the equality comes from Lemma 3.4.5. Thus,

$$\begin{aligned}
\sum_{z=1}^{p} f(\lambda^{T_{z+1}}) - f(\lambda^{T_z}) &\geq \frac{4}{3Ln^4\kappa}\sum_{z=1}^{p}\|\tilde{v}^{T_z}\|^2 \\
f(\lambda^{T_{p+1}}) - f(\lambda^1) &\geq \frac{4}{3Ln^4\kappa}\sum_{z=1}^{p}\|\tilde{v}^{T_z}\|^2 \\
f(\lambda^*) - f(\lambda^1) &\geq \frac{4}{3Ln^4\kappa}\sum_{z=1}^{p}\|\tilde{v}^{T_z}\|^2
\end{aligned}$$

The last inequality together with the fact that $p(\min_{z=1,\ldots,p}\|\tilde{v}^{T_z}\|^2) \leq \sum_{z=1}^{p}\|\tilde{v}^{T_z}\|^2$, proves the third claim.

Proof of 4: If the set of optima of (3.1) is bounded, $\{\lambda : \|\tilde{v}(\lambda)\| \leq C\}$ is a bounded set for some $C > 0$. We conclude that λ^{T_z} has a converging subsequence $\lambda^{T_{z_k}}$. Let $\bar{\lambda}$ be the limit of $\lambda^{T_{z_k}}$. Since $\|\tilde{v}(\cdot)\|$ is a continuous function and $\|\tilde{v}(\lambda^{T_{z_k}})\|$ converges to zero, we conclude that $\tilde{v}(\bar{\lambda}) = 0$ and $\bar{\lambda}$ is optimal. Since f is continuous, we conclude that $\{f(\lambda^{T_{z_k}})\}$ converges to $f(\bar{\lambda}) = f(\lambda^*)$. Since $\{f(\lambda^t)\}$ converges, we conclude that it must converge to $f(\lambda^*)$. □

3.4.2 The Non-differentiable Case

In this section we consider concave objective functions that are not required to be differentiable at all points. We note that the resource allocation equivalent of the approximate LP presents an instance where the objective functions are not differentiable at all points. To further motivate our interest in such functions, consider the following optimization problem:

$$\begin{aligned} \max_{x_i, i=1,\ldots,n} \quad & \frac{1}{n}\sum_{i=1}^{n} g_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \mathbf{A_i} x_i \leq B, \end{aligned} \tag{3.7}$$

where $x_i \in \Re^q$, $\mathbf{A_i} \in \Re^{m \times q}, i = 1, \ldots, n$, $B \in \Re^m$ and $g_i(x_i)$ is a concave function, $i = 1, \ldots, n$. Define $f_i(\lambda_i)$ as the optimal value for the following optimization problem:

$$\begin{aligned} f_i(\lambda_i) = \quad & \max_{x_i \in \Re^q} \quad g_i(x_i) \\ & \text{s.t.} \quad \mathbf{A_i} x_i \leq \lambda_i, \end{aligned} \tag{3.8}$$

With this definition of f_i we see that problem (3.7) is equivalent to problem (3.1). Note that if there are linear constraints that involve only the variables $x_{ij}, j = 1, \ldots, q$ for some i then these constraints could be included directly in the problem defining f_i. Suppose that $f_i(\lambda_i)$ is well-defined and finite for all λ_i. It can then be shown that $f_i(\lambda_i)$ is a concave function. Thus we can potentially apply the decentralized algorithm developed in the previous section for finding an optimal solution to (3.7). However, $f_i(\lambda_i)$ is typically non-differentiable even when $g_i(x_i)$ is. Hence Theorem 3.4.1 does not immediately apply to (3.7) as it relies on the assumption that the objective function is differentiable with Lipschitz continuous gradient. This motivates us to consider cases where f_i, $i = 1, \ldots, n$ are not necessarily differentiable at all points.

In this section, we relax Assumption 3.4.2 and consider the case where f_i, $i = 1, \ldots, n$ are non-differentiable. We introduce a smooth approximation for f_i that is amenable to optimization via stochastic approximations and propose a randomized version of (3.2) to solve the smoothed problem. We show that the new scheme converges to a near-optimal solution of the original problem in a tractable number of iterations.

We assume f_i, $i = 1, \ldots, n$ are concave and differentiable outside a set of measure zero. Denote by $\partial f_i(\lambda_i)$ the set of subgradients of f_i at λ_i. Let $\nabla f_i(\lambda_i)$ be an element chosen arbitrarily from $\partial f_i(\lambda_i)$ for each λ_i. Let $\|\cdot\|_1$ denote the l_1 norm and recall that $\|\cdot\|$ denotes the Euclidean norm. We make the following assumption:

Assumption 3.4.3. *For all i and λ_i, $\sup_{i,\lambda_i}\{\|v\|_1 : v \in \partial f_i(\lambda_i)\} \leq L < \infty$.*

Note that $\sup_{i,\lambda_i}\{\|v\| : v \in \partial f_i(\lambda_i)\} \leq L < \infty$, since $\|v\| \leq \|v\|_1$ for all v. We now consider approximating f_i by a suitable differentiable function. In particular, let

$$\hat{f}_i(\lambda_i) = \mathrm{E}[f_i(\lambda_i + Z_i)]$$

where each $Z_i = (Z_{ij})_{j=1,\ldots,m}$ is a vector of m i.i.d. normal random variables [23] with zero mean and variance equal to

$$\sigma = \frac{\sqrt{2}\epsilon}{\sqrt{\pi \log(m+1)}}$$

where ϵ is a parameter related to the accuracy of the approximation as will be clear from the following lemma. The following lemma shows that $\hat{f}_i$ is a concave and differentiable approximation to f_i and that its gradient $\nabla \hat{f}_i$ can be expressed in terms of ∇f_i.

Lemma 3.4.12. *Let f_i and $\hat{f}_i$ be as given above. Then the following hold:*

1. *$\hat{f}_i$ is concave and differentiable with gradient $\nabla \hat{f}_i(\lambda_i) = \mathrm{E}[\nabla f_i(\lambda_i + Z_i)]$;*
2. *$f_i(\lambda_i) \geq \hat{f}_i(\lambda_i) \geq f_i(\lambda_i) - 2.8\epsilon L$;*
3. *$\|\nabla \hat{f}_i(\lambda_i) - \nabla \hat{f}_i(\bar{\lambda}_i)\| \leq \frac{\sqrt{\log(m+1)}L}{\epsilon}\|\lambda_i - \bar{\lambda}_i\|$.*

Proof of 1: For all $a \in [0, 1]$, we have

$$\begin{aligned}
\hat{f}_i(a\lambda_i + (1-a)\bar{\lambda}_i) &= \mathrm{E}[f_i(a\lambda_i + (1-a)\bar{\lambda}_i + Z_i)] \\
&\geq \mathrm{E}[af_i(\lambda_i + Z_i) + (1-a)f_i(\bar{\lambda}_i + Z_i)] = a\hat{f}_i(\lambda_i) + (1-a)\hat{f}_i(\bar{\lambda}_i)
\end{aligned}$$

It follows that $\hat{f}_i$ is concave. Since f_i is non-differentiable only on a set of measure zero, we have

$$\begin{aligned}
(\nabla f_i(\lambda_i + Z_i))_j &= \lim_{\delta \uparrow 0} \frac{f_i(\lambda_i + Z_i + \delta e_j) - f_i(\lambda_i + Z_i)}{\delta} \\
&= \lim_{\delta \downarrow 0} \frac{f_i(\lambda_i + Z_i + \delta e_j) - f_i(\lambda_i + Z_i)}{\delta}
\end{aligned}$$

with probability 1, where e_j is the vector with all entries equal to zero except for the jth entry, which is equal to one. Hence

$$\begin{aligned}
\lim_{\delta \uparrow 0} \frac{\mathrm{E}[f_i(\lambda_i + Z_i + \delta e_j)] - \mathrm{E}[f_i(\lambda_i + Z_i)]}{\delta} &= \mathrm{E}\left[\lim_{\delta \uparrow 0} \frac{f_i(\lambda_i + Z_i + \delta e_j) - f_i(\lambda_i + Z_i)}{\delta}\right] \\
&= \mathrm{E}[(\nabla f_i(\lambda_i + Z_i))_j] \\
&= \mathrm{E}\left[\lim_{\delta \downarrow 0} \frac{f_i(\lambda_i + Z_i + \delta e_j) - f_i(\lambda_i + Z_i)}{\delta}\right] \\
&= \lim_{\delta \downarrow 0} \frac{\mathrm{E}[f_i(\lambda_i + Z_i + \delta e_j)] - \mathrm{E}[f_i(\lambda_i + Z_i)]}{\delta}
\end{aligned}$$

Note that $\left|\frac{f_i(\lambda_i + Z_i + \delta e_j) - f_i(\lambda_i + Z_i)}{\delta}\right| \leq L$. Hence the exchanges between limit and expectation are valid, by the bounded convergence theorem. It follows that $\hat{f}_i$ is differentiable and its gradient is given by

$$\nabla \hat{f}_i(\lambda_i) = \mathrm{E}[\nabla f_i(\lambda_i + Z_i)]$$

Proof of 2: First, we have

$$\begin{aligned}\hat{f}_i(\lambda_i) &= \mathrm{E}[f_i(\lambda_i + Z_i)] \\ &\leq f_i(\lambda_i + EZ_i) = f_i(\lambda_i),\end{aligned}$$

where the inequality follows from concavity of f_i and Jensen's inequality [23].

For the lower bound on $\hat{f}_i$, we have

$$\begin{aligned}\hat{f}_i(\lambda_i) &= \mathrm{E}[f_i(\lambda_i + Z_i)] \\ &= \mathrm{E}[f_i(\lambda_i - Z_i)] \\ &\geq \mathrm{E}[f_i(\lambda_i) - Z_i^T \nabla f_i(\lambda_i - Z_i)] \\ &\geq f_i(\lambda_i) - \mathrm{E}[\max_j |Z_{ij}|]L, \end{aligned} \tag{3.9}$$

where $|Z_{ij}|$ is the modulus function. The first inequality follows from concavity of f and the fact that $\nabla f_i(\lambda_i - Z_i)$ is a subgradient of f at $\lambda_i - Z_i$. The second inequality follows from the fact that $||\nabla f_i(\lambda_i - Z_i)||_1 \leq L$.

We now show that $\mathrm{E}[\max_j |Z_{ij}|] \leq 2.8\epsilon$. Note that this inequality and (3.9) prove the claim.

We first place a bound on $P(|Z_{ij}| > c)$, for $c > 0$. We have

$$\begin{aligned}P(|Z_{ij}| > c) &= \int_c^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}} dz + \int_{-\infty}^{-c} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}} dz \quad = \quad 2\int_c^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}} dz \\ &= 2e^{-\frac{c^2}{2\sigma^2}} \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2+2zc}{2\sigma^2}} dz \\ &= 2e^{-\frac{c^2}{2\sigma^2}} \left(\int_0^c \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2+2zc}{2\sigma^2}} dz + \int_c^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2+2zc}{2\sigma^2}} dz \right) \\ &\leq 2e^{-\frac{c^2}{2\sigma^2}} \left(\int_0^c \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{zc}{\sigma^2}} dz + \int_c^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}} dz \right) \\ &= 2e^{-\frac{c^2}{2\sigma^2}} \left(\frac{\sigma}{\sqrt{2\pi}c} \left(1 - e^{-\frac{c^2}{\sigma^2}}\right) + \frac{1}{2} P(|Z_{ij}| > c) \right)\end{aligned}$$

Hence

$$
\begin{aligned}
P(|Z_{ij}| > c) &\leq 2\left(\frac{\frac{e^{-\frac{c^2}{2\sigma^2}}\sigma}{\sqrt{2\pi}c}\left(1 - e^{-\frac{c^2}{\sigma^2}}\right)}{1 - e^{-\frac{c^2}{2\sigma^2}}}\right) \\
&= \frac{2e^{-\frac{c^2}{2\sigma^2}}\sigma}{\sqrt{2\pi}c}\left(1 + e^{-\frac{c^2}{2\sigma^2}}\right) \leq \frac{2e^{-\frac{c^2}{2\sigma^2}}\sqrt{2}\sigma}{\sqrt{\pi}c}
\end{aligned}
$$

It follows that, for all $c \geq \frac{2\epsilon}{\sqrt{\pi}}$,

$$
\begin{aligned}
P(\max_j |Z_{ij}| > c) &\leq 2m\frac{e^{-\frac{c^2}{2\sigma^2}}\sqrt{2}\sigma}{\sqrt{\pi}c} \quad \leq \quad 2(m+1)\frac{e^{-\frac{c^2\pi\log(m+1)}{4\epsilon^2}}}{\sqrt{\pi\log(m+1)}} \\
&= \frac{2e^{-\frac{\left(c^2 - \frac{4\epsilon^2}{\pi}\right)\pi\log(m+1)}{4\epsilon^2}}}{\sqrt{\pi\log(m+1)}} \quad \leq \quad \frac{2e^{-\frac{\left(c - \frac{2\epsilon}{\sqrt{\pi}}\right)^2\pi\log(m+1)}{4\epsilon^2}}}{\sqrt{\pi\log(m+1)}}
\end{aligned}
$$

The first inequality follows from the union bound [23]. The second inequality follows from $c \geq \frac{2\epsilon}{\sqrt{\pi}}$. The last inequality follows from $\left(c - \frac{2\epsilon}{\sqrt{\pi}}\right)^2 \leq c^2 - \frac{4\epsilon^2}{\pi}$ for all $c > \frac{2\epsilon}{\sqrt{\pi}}$.

Finally,

$$
\begin{aligned}
\mathrm{E}[\max_j |Z_{ij}|] &= \int_0^\infty P(\max_j |Z_{ij}| > z)dz \\
&= \int_0^{\frac{2\epsilon}{\sqrt{\pi}}} P(\max_j |Z_{ij}| > z)dz + \int_{\frac{2\epsilon}{\sqrt{\pi}}}^\infty P(\max_j |Z_{ij}| > z)dz \\
&\leq \frac{2\epsilon}{\sqrt{\pi}} + \int_{\frac{2\epsilon}{\sqrt{\pi}}}^\infty \frac{2e^{-\frac{\left(z-\frac{2\epsilon}{\sqrt{\pi}}\right)^2 \pi \log(m+1)}{4\epsilon^2}}}{\sqrt{\pi \log(m+1)}} dz \\
&= \frac{2\epsilon}{\sqrt{\pi}} + \frac{4\epsilon}{\sqrt{\pi}\log(m+1)} \int_{\frac{2\epsilon}{\sqrt{\pi}}}^\infty \frac{\sqrt{\log(m+1)}}{2\epsilon} e^{-\frac{\left(z-\frac{2\epsilon}{\sqrt{\pi}}\right)^2 \pi \log(m+1)}{4\epsilon^2}} dz \\
&= \frac{2\epsilon}{\sqrt{\pi}} + \frac{4\epsilon}{\sqrt{\pi}\log(m+1)} \int_{\frac{2\epsilon}{\sqrt{\pi}}}^\infty \frac{1}{\sqrt{2\pi}} \frac{\sqrt{\pi \log(m+1)}}{\sqrt{2}\epsilon} e^{-\frac{\left(\frac{\sqrt{\pi \log(m+1)}\left(z-\frac{2\epsilon}{\sqrt{\pi}}\right)}{\sqrt{2}\epsilon}\right)^2}{2}} dz \\
&= \frac{2\epsilon}{\sqrt{\pi}} + \frac{2\epsilon}{\sqrt{\pi}\log(m+1)} \\
&\leq 2.8\epsilon
\end{aligned}
$$

The last equality comes from the identity, $\frac{1}{\sqrt{2\pi}}\int_0^\infty e^{-\frac{t^2}{2}}dt = \frac{1}{2}$.

Proof of 3: Denote by $p(\cdot)$ the probability density function for Z_i, i.e., the joint probability density function for $Z_{i1}, \ldots, Z_{im}$. Then we have

$$
\begin{aligned}
\nabla \hat{f}_i(\lambda_i) &= \int_{\Re^m} p(z) \nabla f_i(\lambda_i + z) dz \\
&= \int_{\Re^m} p(z - \lambda_i) \nabla f_i(z) dz
\end{aligned}
$$

It follows that

$$
\begin{aligned}
\|\nabla \hat{f}_i(\lambda_i) - \nabla \hat{f}_i(\bar{\lambda}_i)\| &= \left\| \int_{\Re^m} (p(z-\lambda_i) - p(z-\bar{\lambda}_i)) \nabla f_i(z) dz \right\| \\
&\leq \int_{\Re^m} |p(z-\lambda_i) - p(z-\bar{\lambda}_i)| \|\nabla f_i(z)\| dz \\
&\leq L \int_{\Re^m} |p(z-\lambda_i) - p(z-\bar{\lambda}_i)| dz \qquad (3.10)
\end{aligned}
$$

Since $p(\cdot)$ is the joint distribution of m i.i.d., zero-mean Gaussian random variables, $p(z)$ is strictly decreasing on $\|z\|$. Hence

$$
\begin{aligned}
\int_{\Re^m} |p(z-\lambda_i) - p(z-\bar{\lambda}_i)| dz &= \int_{\{z \in \Re^m : \|z-\lambda_i\| < \|z-\bar{\lambda}_i\|\}} \big(p(z-\lambda_i) - p(z-\bar{\lambda}_i)\big)\, dz + \\
&\quad + \int_{\{z \in \Re^m : \|z-\lambda_i\| > \|z-\bar{\lambda}_i\|\}} \big(p(z-\bar{\lambda}_i) - p(z-\lambda_i)\big)\, dz \\
&= 2 \int_{\{z \in \Re^m : \|z-\lambda_i\| < \|z-\bar{\lambda}_i\|\}} \big(p(z-\lambda_i) - p(z-\bar{\lambda}_i)\big)\, dz \\
&= 2 \int_{\{z \in \Re^m : \|z\| < \|z-(\bar{\lambda}_i - \lambda_i)\|\}} p(z) dz - 2 \int_{\{z \in \Re^m : \|z\| > \|z-(\lambda_i - \bar{\lambda}_i)\|\}} p(z) dz \\
&= 2P(\|Z_i\| < \|Z_i - (\bar{\lambda}_i - \lambda_i)\|) - 2P(\|Z_i\| > \|Z_i - (\lambda_i - \bar{\lambda}_i)\|) \\
&= 2P(2Z_i^T(\bar{\lambda}_i - \lambda_i) < \|\bar{\lambda}_i - \lambda_i\|^2) - 2P(2Z_i^T(\bar{\lambda}_i - \lambda_i) < -\|\bar{\lambda}_i - \lambda_i\|^2) \\
&= 2P(-0.5\|\bar{\lambda}_i - \lambda_i\| < V < 0.5\|\bar{\lambda}_i - \lambda_i\|), \qquad (3.11)
\end{aligned}
$$

where

$$
V = \frac{Z_i^T(\bar{\lambda}_i - \lambda_i)}{\|\bar{\lambda}_i - \lambda_i\|}
$$

It is easy to verify that V is normal with zero mean and variance equal to $\sigma = \frac{\sqrt{2}\epsilon}{\sqrt{\pi \log(m+1)}}$. It follows that

$$
\begin{aligned}
P(-0.5\|\bar{\lambda}_i - \lambda_i\| < V < 0.5\|\bar{\lambda}_i - \lambda_i\|) &\leq \frac{1}{\sqrt{2\pi}\sigma} \|\bar{\lambda}_i - \lambda_i\| \\
&= \frac{\sqrt{\log(m+1)}}{2\epsilon} \|\bar{\lambda}_i - \lambda_i\| \qquad (3.12)
\end{aligned}
$$

The claim follows from (3.10), (3.11) and (3.12). $\square$

Bearing in mind the previous lemma, we consider the problem of maximizing

$$\max_\lambda \quad \hat{f}(\lambda) = \sum_{i=1}^{n} \frac{1}{n} \hat{f}_i(\lambda_i) \tag{3.13}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \lambda_i = B$$

Since $\hat{f}_i$ is differentiable with Lipschitz continuous gradient, Theorem 3.4.1 ensures that the update rule (3.2) leads to convergence. However, note that computing the gradient of $\hat{f}_i$ requires evaluating the expected value $\nabla \hat{f}_i(\lambda_i) = \mathrm{E}[\nabla f_i(\lambda_i + Z_i)]$, which is in general computationally expensive. Due to the special form of the smoothing scheme and, in particular, the fact that $\nabla \hat{f}_i$ is expressed as the expected value of the subgradient of f_i, we consider instead of (3.2) a stochastic approximation version of the update. In particular, we let

$$\lambda_i^{t+1} = \lambda_i^t + \gamma_t \sum_{j \in N_i(t)} \frac{1}{n} (\nabla f_i(\lambda_i^t + Z_i^t) - \nabla f_j(\lambda_j^t + Z_j^t)), \tag{3.14}$$

where Z_i^t, $t = 1, 2, \ldots$ is a sequence of i.i.d. vectors with the same distribution as Z_i.

For each λ, let $\tilde{v}(\lambda)$ be given by

$$\tilde{v}_i(\lambda) = \sum_{j \in N} \frac{1}{n} (\nabla \hat{f}_i(\lambda_i) - \nabla \hat{f}_j(\lambda_j))$$

Let $\tilde{v}^t = \tilde{v}(\lambda^t)$ and note that $\tilde{v}^t$ corresponds to the expected direction of update, when the communication graph is complete. From Lemma 3.4.1, it is clear that a feasible solution λ is optimal for (3.13) if and only if $||\tilde{v}(\lambda)|| = 0$. Furthermore, from Lemma 3.4.12, if λ is optimal for (3.13), then it is also near-optimal for (3.1). The following theorem establishes that, if all agents apply (3.14), then $||\tilde{v}^t||$ converges to zero.

We make the following assumption on the stepsizes γ_t:

Assumption 3.4.4. *The stepsizes γ_t satisfy $\gamma_t = \frac{\epsilon}{(2L\sqrt{\log(m+1)})}\beta_t$, where $0 \leq \beta_{t+1} \leq \beta_t \leq 1 \forall t$, $\sum_t \beta_t = \infty$ and $\sum_t \beta_t^2 < \infty$.*

Theorem 3.4.2. *Suppose that Assumptions 3.4.3 and 3.4.4 hold. Then with probability 1:*

1. *The sequence $\{\|\tilde{v}^{T_z}\|\}$ converges to 0.*
2. $\min_{z=1,\dots,p} \mathrm{E}[\|\tilde{v}^{T_z}\|^2] \leq \frac{\frac{n^4 \kappa L \sqrt{\log(m+1)}}{\epsilon}\left[3(f(\lambda^*)-f(\lambda^1)+2.8\epsilon L)+\sum_{t=1}^{t=\kappa p} \frac{4L\beta_t^2\epsilon}{\sqrt{\log(m+1)}}\right]}{4\sum_{z=2}^{p+1}\beta_{\kappa z}}, \forall p$
3. *If the set of optima of (3.1) is bounded, then* $\lim_{t\to\infty} f(\lambda^t) \geq f(\lambda^*) - 2.8\epsilon L$.

The proof has the same structure as the proof of Theorem 3.4.1. Let the expected direction of update at time t be v^t:

$$v_i^t = \frac{1}{n}\sum_{j\in N_i(t)} (\nabla \hat{f}_i(\lambda_i^t) - \nabla \hat{f}_j(\lambda_j^t))$$

Let δ^t be the random variable denoting the difference between the actual and expected directions of update:

$$\delta_i^t = \sum_{j\in N_i(t)} \frac{1}{n}(\nabla f_i(\lambda_i^t + Z_i^t) - \nabla f_j(\lambda_j^t + Z_j^t)) - v_i^t$$

Let $\mathcal{F}_t$ be the sigma-algebra [23] generated by Z_i^τ, $i = 1,\dots,n, \tau = 1,\dots,t$. We have the following result about δ_i^t.

Lemma 3.4.13. *For all t, $\mathrm{E}[\delta^t|\mathcal{F}_{t-1}] = 0$ and $\mathrm{E}[\|\delta^t\|^2|\mathcal{F}_{t-1}] < 8nL^2$, with probability 1.*

Proof. $\mathrm{E}[\delta_i^t|\mathcal{F}_{t-1}] = 0$ follows from $\nabla\hat{f}_i(\lambda_i) = \mathrm{E}[\nabla f_i(\lambda_i + Z_i^t)]$ for all i. Moreover,

$$\begin{aligned}
\mathrm{E}[\|\delta_i^t\|^2|\mathcal{F}_{t-1}] &= \mathrm{E}\left[\left\|\sum_{j\in N_i(t)} \frac{\nabla f_i(\lambda_i^t + Z_i^t) - \nabla\hat{f}_i(\lambda_i^t) - \nabla f_j(\lambda_j^t + Z_j^t) + \nabla\hat{f}_j(\lambda_j^t)}{n}\right\|^2 |\mathcal{F}_{t-1}\right] \\
&= \frac{\mathrm{E}\left[\left\|N_i(t)\left(\nabla f_i(\lambda_i^t + Z_i^t) - \nabla\hat{f}_i(\lambda_i^t)\right) - \sum_{j\in N_i(t)}\left(\nabla f_j(\lambda_j^t + Z_j^t) - \nabla\hat{f}_j(\lambda_j^t)\right)\right\|^2 |\mathcal{F}_{t-1}\right]}{n^2} \\
&\leq \frac{N_i(t)^2\mathrm{E}[\|\nabla f_i(\lambda_i^t + Z_i^t) - \nabla\hat{f}_i(\lambda_i^t)\|^2|\mathcal{F}_{t-1}] + \sum_{j\in N_i(t)}\mathrm{E}[\|\nabla f_j(\lambda_j^t + Z_j^t) - \nabla\hat{f}_j(\lambda_j^t)\|^2|\mathcal{F}_t}{n^2} \\
&< 8L^2
\end{aligned}$$

The last inequality follows from $N_i(t) < n$ and

$$\|\nabla f_j(\lambda_j^t + Z_j^t) - \nabla \hat{f}_j(\lambda_j^t)\| \;\leq\; \|\nabla f_j(\lambda_j^t + Z_j^t)\| + \|\nabla \hat{f}_j(\lambda_j^t)\| \leq 2L$$

Finally,

$$\mathrm{E}[\|\delta^t\|^2|\mathcal{F}_{t-1}] \;=\; \sum_i \mathrm{E}[\|\delta_i^t\|^2|\mathcal{F}_{t-1}] < 8nL^2$$

□

The following results follow immediately from Lemmas 3.4.3-3.4.6 applied with $\hat{f}_i$ replacing f_i for all i:

$$\nabla \hat{f}(\lambda^t)^T v^t = \frac{1}{n^2} \sum_{(i,j)\in E(t)} \|\nabla \hat{f}_i(\lambda_i^t) - \nabla \hat{f}_j(\lambda_j^t)\|^2 \tag{3.15}$$

$$\|v^t\|^2 \leq 2n \nabla \hat{f}(\lambda^t)^T v^t \tag{3.16}$$

$$\|\tilde{v}^t\|^2 = n \nabla \hat{f}(\lambda^t)^T \tilde{v}^t \tag{3.17}$$

$$\sum_{(i,j)\in E} \|\nabla \hat{f}_j(\lambda_j^t) - \nabla \hat{f}_i(\lambda_i^t)\|^2 \geq \frac{8}{n^3} \sum_{((i,j)\in N^2, i<j)} \|\nabla \hat{f}_j(\lambda_j^t) - \nabla \hat{f}_i(\lambda_i^t)\|^2$$
$$\forall E : (N, E) \text{ is connected.} \tag{3.18}$$

Let E_{T_z} be a subset of the edge set $E_{T_z,T_{z+1}}$ such that the graph (N, E_{T_z}) is a tree. By Assumption 3.3.1, the graph $(N, E_{T_z,T_{z+1}})$ is connected and so E_{T_z} is well defined. As before, let the decentralized direction of update derived using $G = (N, E_{T_z})$ be denoted by $\bar{v}^{T_z}$. The following result is the counterpart of Lemma 3.4.11.

Lemma 3.4.14. *Let* $L_\epsilon = \frac{\sqrt{\log(m+1)}L}{\epsilon}$.

$$\nabla \hat{f}(\lambda^{T_z})^T \bar{v}^{T_z} \leq \kappa \sum_{t=T_z}^{t=T_{z+1}-1} \left[(1 + 2L_\epsilon^2\gamma_t^2)\mathrm{E}[\nabla \hat{f}(\lambda^t)^T v^t|\mathcal{F}_{T_z-1}] + 8L^2L_\epsilon^2\gamma_t^2\right].$$

From the Cauchy-Schwarz inequality,

$$\begin{aligned}
\|w_{ij}(T_z)\|^2 &\leq \frac{(t^i_{T_z} - T_z + 1)}{n^2} \Big(\|\mathrm{E}[\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}}) | \mathcal{F}_{T_z - 1}]\|^2 + \\
&\quad + \sum_{t=T_z}^{t=t^i_{T_z}-1} \|\mathrm{E}[\nabla \hat{f}_j(\lambda_j^{t+1}) - \nabla \hat{f}_j(\lambda_j^t) | \mathcal{F}_{T_z - 1}]\|^2 \Big) \\
&\leq \frac{\kappa}{n^2} \left(\mathrm{E}[\|\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}})\|^2 | \mathcal{F}_{T_z-1}] + \sum_{t=T_z}^{t=t^i_{T_z}-1} \mathrm{E}[\|\nabla \hat{f}_j(\lambda_j^{t+1}) - \nabla \hat{f}_j(\lambda_j^t)\|^2) | \mathcal{F}_{T_z-1}] \right) \\
&\leq \kappa \left(\frac{\mathrm{E}[\|\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}})\|^2 | \mathcal{F}_{T_z-1}]}{n^2} + \frac{L_\epsilon^2}{n^2} \sum_{t=T_z}^{t=T_{z+1}-1} \gamma_t^2 (\mathrm{E}[2n \nabla \hat{f}(\lambda^t)^T v^t | \mathcal{F}_{T_z-1}] + 8nL^2) \right.
\end{aligned}$$

The last inequality follows from the fact that $\|\nabla \hat{f}(\lambda^{t+1}) - \nabla \hat{f}(\lambda^t)\|^2 = \frac{1}{n^2} \sum_{k=1}^n \|\nabla \hat{f}_k(\lambda_k^{t+1}) - \nabla \hat{f}_k(\lambda_k^t)\|^2 \geq \frac{1}{n^2} \|\nabla \hat{f}_j(\lambda_j^{t+1}) - \nabla \hat{f}_j(\lambda_j^t)\|^2$ and from (3.19). We finally have

$$\begin{aligned}
\nabla \hat{f}(\lambda^{T_z})^T \bar{v}^{T_z} &= \sum_{(i,j) \in E_{T_z}} \|w_{ij}(T_z)\|^2 \\
&\leq \sum_{(i,j) \in E_{T_z}} \kappa \left(\frac{\mathrm{E}[\|\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}})\|^2 | \mathcal{F}_{T_z-1}]}{n^2} + \right. \\
&\quad \left. + \frac{L_\epsilon^2}{n^2} \sum_{t=T_z}^{t=T_{z+1}-1} \gamma_t^2 (\mathrm{E}[2n \nabla \hat{f}(\lambda^t)^T v^t | \mathcal{F}_{T_z-1}] + 8nL^2) \right) \\
&\leq \kappa \sum_{t=T_z}^{t=T_{z+1}-1} \left[(1 + 2L_\epsilon^2 \gamma_t^2) \mathrm{E}[\nabla \hat{f}(\lambda^t)^T v^t | \mathcal{F}_{T_z-1}] + 8L^2 L_\epsilon^2 \gamma_t^2 \right]
\end{aligned}$$

In the last inequality, we have used the fact that

$$\begin{aligned}
\sum_{(i,j) \in E_{T_z}} \frac{\mathrm{E}[\|\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}})\|^2 | \mathcal{F}_{T_z-1}]}{n^2} &= \sum_{t=T_z}^{t=T_{z+1}-1} \sum_{(i,j) \in E_{T_z}, t^i_{T_z} = t} \frac{\mathrm{E}[\|\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}})\|^2 | \mathcal{F}_{T_z-1}]}{n^2} \\
&\leq \sum_{t=T_z}^{t=T_{z+1}-1} \mathrm{E}[\nabla \hat{f}(\lambda^t)^T v^t | \mathcal{F}_{T_z-1}]
\end{aligned}$$

Proof. We have

$$\begin{aligned}
\mathrm{E}[\|\nabla \hat{f}(\lambda^{t+1}) - \nabla \hat{f}(\lambda^t)\|^2 | \mathcal{F}_{t-1}] &\leq \frac{L_\epsilon^2}{n^2}\mathrm{E}[\|\gamma_t(v^t + \delta^t)\|^2 | \mathcal{F}_{t-1}] \\
&= \frac{L_\epsilon^2 \gamma_t^2}{n^2}(\|v^t\|^2 + \mathrm{E}[\|\delta^t\|^2 | \mathcal{F}_{t-1}]) \\
&\leq \frac{L_\epsilon^2 \gamma_t^2}{n^2}(2n\nabla \hat{f}(\lambda^t)^T v^t + 8nL^2) \qquad (3.19)
\end{aligned}$$

It follows from Lemma 3.4.12 that L_ϵ is a Lipschitz constant for the functions $\hat{f}_i$, $i = 1, \ldots, n$. Hence $\frac{L_\epsilon}{n}$ is a Lipschitz constant for $\hat{f}$ and the first inequality follows from this. The second inequality follows from (3.16) and Lemma 3.4.13.

Let $t^i_{T_z}$ be the earliest time between the time periods T_z and $T_{z+1} - 1$ such that there is an edge $(i, j) \in E_{T_z}$ for agent i. It is clear that $T_z \leq t^i_{T_z} \leq T_{z+1} - 1$. Also, by definition, for $l = T_z, T_z + 1, \ldots, (t^i_{T_z} - 1)$, there is no edge $(i, p) \in E(l)$. Thus $\lambda_i^{t^i_{T_z}} = \lambda_i^{T_z}$ and $\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) = \nabla \hat{f}_i(\lambda_i^{T_z})$. Let $w_{ij}(t) = \frac{1}{n}(\nabla \hat{f}_i(\lambda_i^t) - \nabla \hat{f}_j(\lambda_j^t))$. Then

$$\begin{aligned}
\|w_{ij}(T_z)\| &= \frac{1}{n}\|\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{T_z})\| \\
&\leq \frac{1}{n}(\|\mathrm{E}[\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}}) | \mathcal{F}_{T_z - 1}]\| + \|\mathrm{E}[\nabla \hat{f}_j(\lambda_j^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{T_z}) | \mathcal{F}_{T_z - 1}]\|) \\
&\leq \frac{1}{n}(\|\mathrm{E}[\nabla \hat{f}_i(\lambda_i^{t^i_{T_z}}) - \nabla \hat{f}_j(\lambda_j^{t^i_{T_z}}) | \mathcal{F}_{T_z - 1}]\| + \sum_{t=T_z}^{t=t^i_{T_z}-1} \|\mathrm{E}[\nabla \hat{f}_j(\lambda_j^{t+1}) - \nabla \hat{f}_j(\lambda_j^t) | \mathcal{F}_{T_z - 1}]\|)
\end{aligned}$$

□

Proof of theorem 3.4.2: Proof of 1: We first have

$$\begin{aligned}
\mathrm{E}[\hat{f}(\lambda^{t+1})|\mathcal{F}_{t-1}] &\geq \hat{f}(\lambda^t) + \gamma_t \nabla \hat{f}(\lambda^t)^T v^t - \frac{L_\epsilon}{2n}\mathrm{E}[\|\gamma_t(v^t+\delta^t)\|^2|\mathcal{F}_{t-1}] \\
&= \hat{f}(\lambda^t) + \gamma_t \nabla \hat{f}(\lambda^t)^T v^t - \frac{L_\epsilon}{2n}\|\gamma_t v^t\|^2 - \frac{L_\epsilon}{2n}\mathrm{E}[\|\gamma_t \delta^t\|^2|\mathcal{F}_{t-1}] \\
&\geq \hat{f}(\lambda^t) + (\gamma_t - L_\epsilon \gamma_t^2)\nabla \hat{f}(\lambda^t)^T v^t - 4L_\epsilon L^2 \gamma_t^2 \qquad (3.20)
\end{aligned}$$

The first inequality comes from the descent lemma for differentiable functions [4]. The equality follows from $\mathrm{E}[\delta|\mathcal{F}_{t-1}] = 0$, from Lemma 3.4.13. The second inequality follows from Lemma 3.4.13 and (3.16).

Note that $\nabla \hat{f}(\lambda^t)^T v^t \geq 0$. This and Assumption 3.4.4 imply that the second term in (3.20) is also greater than or equal to zero. Moreover,

$$\sum_t 4L_\epsilon L^2 \gamma_t^2 < \infty$$

Since $\hat{f}$ is bounded from above, we conclude by the supermartingale convergence theorem [23] that $\hat{f}(\lambda^t)$ converges with probability 1. Moreover, $\sum_t(\gamma_t - L_\epsilon\gamma_t^2)\nabla \hat{f}(\lambda^t)^T v^t < \infty$ with probability 1, and since $\sum_t \gamma_t = \infty$, we conclude that $\nabla \hat{f}(\lambda^t)^T v^t$ converges to zero with probability 1. Note that

$$\mathrm{E}[\nabla \hat{f}(\lambda^t)^T v^t|\mathcal{F}_{t-1}] = \nabla \hat{f}(\lambda^t)^T v^t,$$

with probability 1 and we conclude that $\mathrm{E}[\nabla \hat{f}(\lambda^t)^T v_t|\mathcal{F}_{t-1}]$ also converges to zero with probability 1.

We now have

$$\begin{aligned}
\|\tilde{v}^{T_z}\|^2 &= n\nabla\hat{f}(\lambda^{T_z})^T\tilde{v}^{T_z} \\
&\leq \frac{n^4}{8}\nabla\hat{f}(\lambda^{T_z})^T\tilde{v}^{T_z} \\
&\leq \frac{n^4}{8}\kappa\sum_{t=T_z}^{t=T_{z+1}-1}\left[(1+2L_\epsilon^2\gamma_t^2)\mathrm{E}[\nabla\hat{f}(\lambda^t)^T v^t|\mathcal{F}_{T_z-1}] + 8L^2L_\epsilon^2\gamma_t^2\right] \\
&\leq \frac{n^4}{8}\kappa\sum_{t=T_z}^{t=T_{z+1}-1}\left[1.5\mathrm{E}[\nabla\hat{f}(\lambda^t)^T v^t|\mathcal{F}_{T_z-1}] + 2L^2\beta_t^2\right] \qquad (3.21)
\end{aligned}$$

The equality follows from (3.17). The first inequality follows from (3.18) and (3.15). The second inequality follows from Lemma 3.4.14. The third inequality follows from Assumption 3.4.4 on the stepsizes γ_t. We conclude that $\|\tilde{v}^{T_z}\|$ converges to zero with probability 1.

Proof of 2: From (3.20), we have

$$\begin{aligned}
\nabla\hat{f}(\lambda^t)^T v^t &\leq \frac{\mathrm{E}[\hat{f}(\lambda^{t+1})|\mathcal{F}_{t-1}] + 4L_\epsilon L^2\gamma_t^2 - \hat{f}(\lambda^t)}{\gamma_t(1-L_\epsilon\gamma_t)} \\
&\leq \frac{2(\mathrm{E}[\hat{f}(\lambda^{t+1})|\mathcal{F}_{t-1}] - \hat{f}(\lambda^t)) + (\frac{2L^2\beta_t^2}{L_\epsilon})}{\gamma_t} \qquad (3.22)
\end{aligned}$$

In the second inequality we have used $\gamma_t \leq \frac{1}{2L_\epsilon}$, from Assumption 3.4.4.

Combining (3.21) and (3.22), we have

$$\begin{aligned}
\mathrm{E}[\|\tilde{v}^{T_z}\|^2] &\leq \frac{n^4}{8}\kappa\sum_{t=T_z}^{t=T_{z+1}-1}\left(\frac{3\mathrm{E}[\hat{f}(\lambda^{t+1}) - \hat{f}(\lambda^t)] + (\frac{3L^2\beta_t^2}{L_\epsilon})}{\gamma_t} + 2L^2\beta_t^2\right) \\
&\leq \frac{n^4}{8\gamma_{\kappa z}}\kappa\sum_{t=T_z}^{t=T_{z+1}-1}\left[3\mathrm{E}[\hat{f}(\lambda^{t+1}) - \hat{f}(\lambda^t)] + \frac{3L^2\beta_t^2}{L_\epsilon} + \frac{L^2\beta_t^2}{L_\epsilon}\right]
\end{aligned}$$

(3.23)

The last inequality follows from Assumption 3.4.4 on the stepsizes. It follows that

$$\begin{aligned}\sum_{z=1}^{p} \gamma_{\kappa z} \mathrm{E}[\|\tilde{v}^{T_z}\|^2] &\leq \frac{n^4}{8}\kappa \sum_{t=1}^{t=T_{p+1}-1} \left[3\mathrm{E}[\hat{f}(\lambda^{t+1}) - \hat{f}(\lambda^t)] + \frac{4L^2\beta_t^2}{L_\epsilon}\right] \\ &\leq \frac{n^4}{8}\kappa \left[3(\hat{f}(\hat{\lambda}) - \hat{f}(\lambda^1)) + \sum_{t=1}^{t=\kappa p} \frac{4L^2\beta_t^2}{L_\epsilon}\right],\end{aligned}$$

where $\hat{\lambda}$ denotes an optimal solution of (3.13). From Lemma 3.4.12, we have $\hat{f}(\lambda^1) \geq f(\lambda^1) - 2.8\epsilon L$. We also have $\hat{f}(\hat{\lambda}) \leq f(\hat{\lambda}) \leq f(\lambda^*)$. It follows that

$$\min_{z=1,\ldots,p} \mathrm{E}[\|\tilde{v}^{T_z}\|^2] \leq \frac{\frac{n^4\kappa L\sqrt{\log(m+1)}}{\epsilon}\left[3(f(\lambda^*) - f(\lambda^1) + 2.8\epsilon L) + \sum_{t=1}^{t=\kappa p} \frac{4L\beta_t^2\epsilon}{\sqrt{\log(m+1)}}\right]}{4\sum_{z=2}^{p+1}\beta_{\kappa z}}, \forall p.$$

Proof of 3: Since $f \geq \hat{f} \geq f - 2.8\epsilon L$, if (3.1) has a bounded set of optima, so does (3.13). Recall from the proof of the first claim that $\hat{f}(\lambda^t)$ converges with probability 1. Using the same argument as in the proof of the fourth claim of Theorem 3.4.1, we conclude that $\hat{f}(\lambda^t)$ converges to $\hat{f}(\hat{\lambda})$ with probability 1. We conclude that

$$\begin{aligned}\limsup_{t\to\infty} f(\lambda^t) &\geq \hat{f}(\hat{\lambda}) \\ &\geq \hat{f}(\lambda^*) \\ &\geq f(\lambda^*) - 2.8\epsilon L\end{aligned}$$

The first inequality follows from $f(\lambda^t) \geq \hat{f}(\lambda^t)$ for all t, from Lemma 3.4.12. The second inequality follows from optimality of $\hat{\lambda}$. The third inequality follows from Lemma 3.4.12. □

It is worth noting some aspects of Theorem 3.4.2. Unlike in the differentiable case, we cannot guarantee monotonic increase in the objective function values. Hence the rate of convergence of the sequence $\{\mathrm{E}[\|\tilde{v}^{T_z}\|]\}$ to zero does not have as far-reaching implications as its counterpart in Theorem 3.4.1. Nevertheless, Theorem 3.4.2 ensures convergence to a near-optimal solution with probability 1. Another substantial difference is on the assumption on stepsizes and the corresponding effect on convergence rates. It is easy to see that convergence is ensured if $\beta_t = \frac{1}{t^q}$, for $0.5 < q \leq 1$. The resulting theoretical rate of convergence is clearly dependent on q; When $0.5 < q < 1$, $\frac{1}{x^q}$ is a decreasing function for $x \geq 1$. Hence for $k \geq 1, \frac{1}{k^q} \geq \int_k^{k+1} \frac{1}{x^q} dx$ and so $\sum_{k=1}^{c} \frac{1}{k^q} \geq \int_1^{c+1} \frac{1}{x^q} dx = \frac{(c+1)^{1-q}-1}{1-q}$. Thus the number of iterations needed for $\mathrm{E}[\|\tilde{v}^{T_z}\|^2] \leq \epsilon$ is polynomial in the problem parameters. Similarly, when $q = 1$, $\frac{1}{x^q}$ is just $\frac{1}{x}$ and is a decreasing function as well for $x \geq 1$. Hence for $k \geq 1$, $\frac{1}{k+1} \leq \int_k^{k+1} \frac{1}{x} dx$ and so $\sum_{k=1}^{c} \frac{1}{k} \leq 1 + \int_1^c \frac{1}{x} dx = \log(c) + 1$ and so the number of iterations needed for $\mathrm{E}[\|\tilde{v}^{T_z}\|^2] \leq \epsilon$ is exponential in the problem parameters. As is often observed in stochastic approximation methods, the impact of the choice of step sizes on the speed of convergence of the algorithm is also verified in the numerical experiments.

3.5 Decentralized resource allocation with non-negativity constraints

In this Section, we use the results developed for (3.1) to solve the following resource allocation problem with non-negativity constraints

$$\begin{aligned} \max_{\lambda_i \in \Re^m, i=1,\ldots,n} \quad & f(\lambda) = \frac{1}{n} \sum_{i=1}^{n} f_i(\lambda_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \lambda_i = B, \\ & \lambda_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{3.24}$$

We assume that f_i is concave and differentiable outside a set of measure zero. Also let assumption 3.4.3 hold for f.

We now define $g_i(\lambda_i)$ as follows,

$$g_i(\lambda_i) = f_i(\lambda_i) + \sum_{j=1}^{m} L_g \min(\lambda_{ij}, 0)$$

where $L_g > 2L$. The following lemma shows that the function $g(\lambda)$ satisfies the assumption 3.4.3 necessary for applying the stochastic approximation version of the gradient-descent algorithm developed in Section 3.4.2.

Lemma 3.5.1. *Under assumption 3.4.3 for f,*

1. *For all i, $g_i(\lambda_i)$ is concave and differentiable outside a set of measure zero.*

2. *For all i and λ_i, $\sup_{i,\lambda_i}\{\|v\|_1 : v \in \partial g_i(\lambda_i)\} \leq L_m < \infty$ where $L_m = L + mL_g$.*

Proof of 1: Let $h_j(\lambda_i) = L_g \min(\lambda_{ij}, 0)$. It is clear that h_j is a piecewise linear function. Recall that

$$g_i(\lambda_i) = f_i(\lambda_i) + \sum_{j=1}^{m} h_j(\lambda_i)$$

The concavity of g_i follows from the concavity of f and the functions h_j, $j = 1, \ldots, m$. The points of non-differentiability of f_i form a set of measure zero. The other points of non-differentiability of g_i are points λ_i, where $\lambda_{ij} = 0$ for some j. These points form a set of measure zero. Thus g_i is differentiable outside a set of measure zero.

Proof of 2:

Let $\mathbf{e}_j$ be the vector whose j'th component is 1 and other components are 0. It is clear that for λ_i with $\lambda_{ij} \neq 0$, h_j is differentiable and $\nabla h_j(\lambda_i) = L_g \mathbf{e}_j$ if $\lambda_{ij} < 0$ and $\nabla h_j(\lambda_i) = \mathbf{0}$ if $\lambda_{ij} > 0$ where $\mathbf{0}$ is the m dimensional zero vector. For λ_i with $\lambda_{ij} = 0$, $\partial h_j(\lambda_i)$ consists of vectors of the form $\bar{L}\mathbf{e}_j$ where $0 \leq \bar{L} \leq L_g$. Thus for all j, $\sup_{\lambda_i}\{\|v\|_1 : v \in \partial h_j(\lambda_i)\} = L_g$. It is known from the theory of convex functions that if $u = \sum_{j=1}^{k} u_j$ where u_j, $j = 1, \ldots, k$ are convex functions , then $\partial u(x) = \sum_{j=1}^{k} \partial u_j(x)$. Thus, if $\sup_x\{\|v\|_1 : v \in \partial u_j(x)\} \leq L_j$, then

$\sup_x\{\|v\|_1 : v \in \partial u(x)\} \leq \sum_{j=1}^{k} L_j$. By assumption, $\sup_{\lambda_i}\{\|v\|_1 : v \in \partial f_i(\lambda_i)\} \leq L$. Hence $\sup_{\lambda_i}\{\|v\|_1 : v \in \partial g_i(\lambda_i)\} \leq L + \sum_{i=1}^{m} L_g = L + mL_g$.

It can be noted from definition of g_i that if $\lambda_i \geq 0$, then $g_i(\lambda_i) = f_i(\lambda_i)$. The term $L_g \min(\lambda_{ij}, 0)$ in the above definition can be thought of as a penalty for negative λ_{ij}. This term ensures that solving (3.1) with g has a non-negative optimal solution and is equivalent to solving (3.24) with f.

Lemma 3.5.2. *The set of optimal solutions for (3.1) with g as the objective function is the same as the set of optimal solutions to* (3.24) *with f as the objective function.*

Proof. Without loss of generality assume that $B > 0$. Consider some optimal solution λ^* for (3.24) with f as the objective function and note that $\lambda^* \geq 0$. Suppose there exists some feasible solution $\hat{\lambda}$ to (3.1) with $\hat{\lambda}_{ij} < 0$ for some i, j. We show that $g(\hat{\lambda}) < g(\lambda^*)$. This implies that solving (3.24) with g as the objective function is equivalent to solving (3.1) with g as the objective function. Since $g(\lambda) = f(\lambda)$ when $\lambda \geq 0$, solving (3.24) with g is equivalent to solving (3.24) with f. Thus the set of optimal solutions for (3.1) with g and for (3.24) with f are the same proving the lemma.

Consider the following problem,

$$
\begin{aligned}
\max_{\lambda_p \in \Re^m, p=1,\ldots,n} \quad & g(\lambda) = \frac{1}{n}\sum_{p=1}^{n} g_p(\lambda_p) \\
\text{s.t.} \quad & \sum_{p=1}^{n} \lambda_p = B, \\
& \lambda_p \geq -|\hat{\lambda}_p|, p = 1, \ldots, n
\end{aligned} \tag{3.25}
$$

It can be seen that λ^* and $\hat{\lambda}$ are feasible solutions to (3.25). We now show that $\hat{\lambda}$ cannot be an optimal solution to (3.25). Since $B > 0$, there exists some k such that $\hat{\lambda}_{kj} > 0$. Define $\bar{\lambda}$ so that it differs from $\hat{\lambda}$ only in the ij and kj components as follows

$$\bar{\lambda}_{ij} = \hat{\lambda}_{ij} + \delta$$

$$\bar{\lambda}_{kj} = \hat{\lambda}_{kj} - \delta$$

where we choose a $\delta > 0$ such that $\bar{\lambda}_{kj} > 0$ and $\bar{\lambda}_{ij} < 0$. It is clear that $\bar{\lambda}$ is a feasible solution to (3.25). We now have:

$$\begin{aligned} g_i(\bar{\lambda}_i) &= f_i(\bar{\lambda}_i) + \sum_{l=1}^{m} h_l(\bar{\lambda}_i) \\ &\geq g_i(\hat{\lambda}_i) + \delta\left(L_g + (\nabla f_i(\bar{\lambda}_i))_j\right) \end{aligned}$$

The inequality comes from the concavity of g_i and from the definition of $\bar{\lambda}$. Similarly

$$\begin{aligned} g_k(\bar{\lambda}_k) &= f_k(\bar{\lambda}_k) + \sum_{l=1}^{m} h_l(\bar{\lambda}_k) \\ &\geq g_k(\hat{\lambda}_k) - \delta\left(\nabla f_k(\bar{\lambda}_k)\right)_j \end{aligned}$$

Hence

$$g_i(\bar{\lambda}_i) + g_k(\bar{\lambda}_k) \geq g_i(\hat{\lambda}_i) + g_k(\hat{\lambda}_k) + \delta\left(L_g + (\nabla f_i(\bar{\lambda}_i))_j - (\nabla f_k(\bar{\lambda}_k))_j\right)$$

Since $L_g > 2L$, we can conclude from the above that $g(\bar{\lambda}) > g(\hat{\lambda})$ and hence $\hat{\lambda}$ cannot be an optimal solution to (3.25).

It can be seen from the definition of (3.25) that its feasible set is bounded. Since g is continuous and since the feasible set of (3.25) is bounded and closed it has at least one optimal solution. The above argument establishes the non-optimality of any feasible solution for (3.25) with at least one non-negative component. Since $g(\lambda) = f(\lambda)$ when $\lambda \geq 0$, solving (3.25) with g is equivalent to solving (3.24) with f and hence λ^* is an optimal solution for (3.25). Hence $g(\hat{\lambda}) < g(\bar{\lambda}) \leq g(\lambda^*)$. □

Since the set of feasible solutions of (3.24) is bounded and closed and since f is assumed to be continuous, there exists an optimal solution to (3.24). Lemma (3.5.2) ensures that any algorithm that finds an optimal solution to (3.1) with g as the objective function also yields an optimal solution to (3.24) with f as the objective function.

Lemma 3.5.1 shows that we can apply the stochastic approximation version of the gradient-descent algorithm for (3.1) with g as the objective function. Hence an optimal solution for (3.24) with f as the objective function can be found by applying the stochastic approximation version of the gradient-descent algorithm developed in Section 3.4.2 for (3.1) with g as the objective function. It should be pointed out that the Lipschitz constant of the smoothed problem and consequently the convergence rate is now of the order $O(\frac{m\sqrt{m}}{\epsilon})$ as compared to $O(\frac{\sqrt{m}}{\epsilon})$ for the results of Section 3.4.2.

3.6 Numerical Experiments

In this section, we present results of numerical experiments which illustrate the performance of the algorithms presented in the previous sections. We compare the proposed algorithms to centralized algorithms that use direction $\tilde{v}(\lambda)$ as the direction of update. Recall that $\tilde{v}(\lambda)$ is the direction of update if the current resource allocation is λ and the communication graph is complete. Recall also that when f_i is differentiable, $\tilde{v}(\lambda)$ is the projection of ∇f onto the subspace $\sum_{i=1}^{n} \lambda_i^t = B$. Thus the centralized algorithm reduces to the classic gradient descent method of non-linear optimization in this case. We define $p^t = (\frac{f^t - f^0}{f^* - f^0}) \times 100$ where f^t is the objective function value after t iterations and f^* is the objective function value of the optimal solution and investigate how p^t converges to hundred in the centralized and decentralized algorithms.

3.6.1 Problem with differentiable objective function

We first consider a problem studied in [38], which is an instance of (3.1) with

$$f_i(x_i) = -(\frac{1}{2}a_i(x_i - c_i)^2 + log(1 + e^{b_i(x_i - d_i)})), i = 1, \ldots, n$$

The second derivative $f_i^{''}$ is given by

$$f_i^{''}(x_i) = -(a_i + b_i^2 \frac{e^{b_i(x_i-d_i)}}{(1+e^{b_i(x_i-d_i)})^2}) \qquad i = 1, \ldots, n$$

It can be verified that $f_i^{''}(x_i)$ has a lower bound $-(a_i + \frac{1}{4}b_i^2)$, $i = 1, \ldots, n$. It can be shown that if a one-dimensional function is differentiable and its gradient is bounded by some constant, then the function is Lipschitz continuous with the same constant. Since f_i is twice differentiable and $f_i^{''}$ is bounded, it follows that $f_i^{'}$ is Lipschitz continuous with constant $(a_i + \frac{1}{4}b_i^2)$, if we assume that $a_i \geq 0$. It follows that $f^{'}$ is Lipschitz continuous with constant $\frac{L}{n}$ where $L = max_i \left(a_i + \frac{1}{4}b_i^2\right)$. Thus f satisfies Assumption 3.4.2.

We choose problem instances with 20 agents and as in [38], the coefficients a_i, b_i, c_i and d_i are generated randomly with uniform distributions on $[0, 2], [-2, 2], [-10, 10]$ and $[-10, 10]$ respectively. Recall that for our algorithm to converge, the union of communication graphs should be connected periodically. For a chosen κ, we let the edges $(i, i+1), i = 1, \ldots, n-1$ be part of the communication graph $E(t)$ for some arbitrarily chosen t such that $m\kappa < t \leq (m+1)\kappa, m = 0, 1, \ldots$. This ensures that $G = (N, E_{m\kappa+1, m(\kappa+1)+1})$ is connected (Recall that $E_{k,l} = \cup_{t=k}^{t=l-1} E(t)$). We let every other edge (i, j) with $j \neq i+1$ be a part of at the most one communication graph between iterations $m\kappa + 1$ and $(m+1)\kappa$ with a probability e_p. The parameter e_p controls the density of the graph, $G = (N, E_{m\kappa+1, m(\kappa+1)+1})$. The step size is chosen to be $\frac{1}{2L}$ with L as defined above. Figure 3-1 shows the convergence behavior of the algorithm for various values of the parameter κ with $e_p = 0.1$. p^t in the figure represents the average of p^t for 10 randomly chosen problems. It can be seen from the figure that the performance of the decentralized algorithm is comparable to the centralized algorithm for $\kappa = 1$ even though the communication graph is not dense ($e_p = 0.1$).

Figure 3-2 shows a comparison of the convergence behavior of the algorithms for problems with varying number of agents. We fix $\kappa = 1$ in these problems and $e_p = 0.1$. The other parameters are chosen as described above. We notice from the Figure 3-2 that the scaling of the performance of decentralized algorithms with increasing number of agents is much better than $O(n^4)$ promised by Theorem 3.4.1.

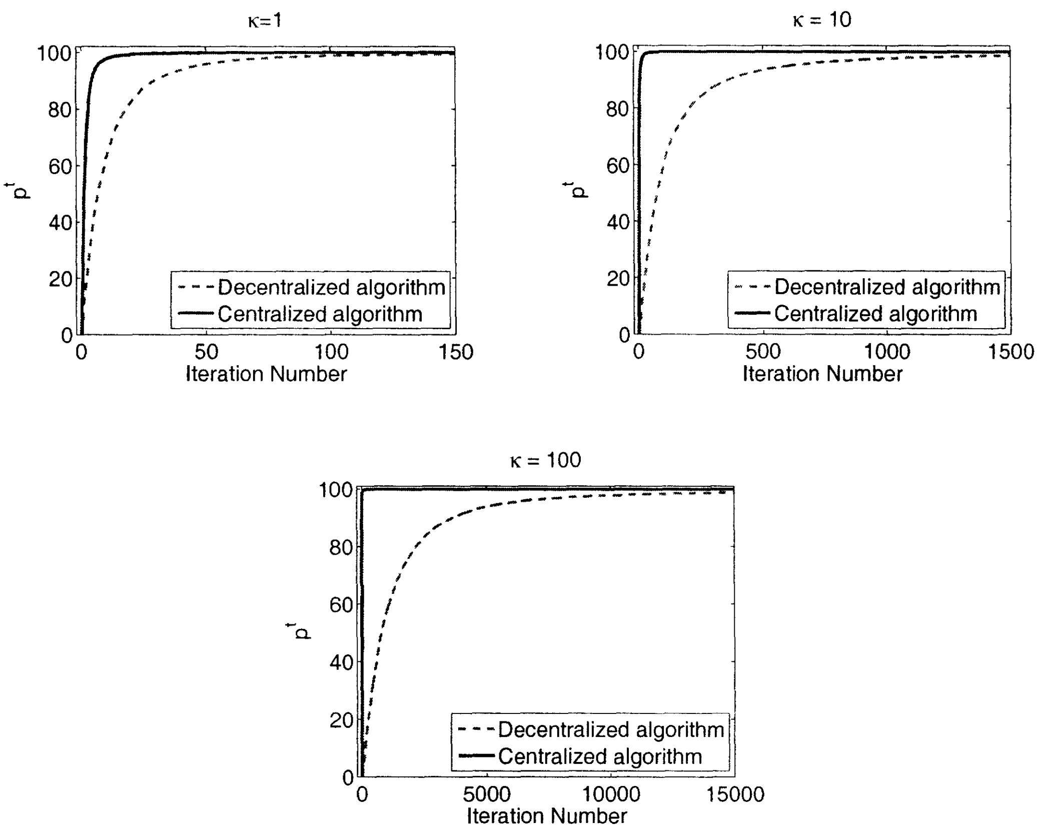

Figure 3-1: Comparison of convergence behavior of the decentralized and centralized algorithms for various κ.

3.6.2 Decentralized optimization of linear programming problems

We now consider decentralized solution of linear programming problems using the randomized version of the decentralized subgradient-descent algorithm developed in this Chapter. We note that the approximate LP is an example of a linear programming problem that requires a decentralized

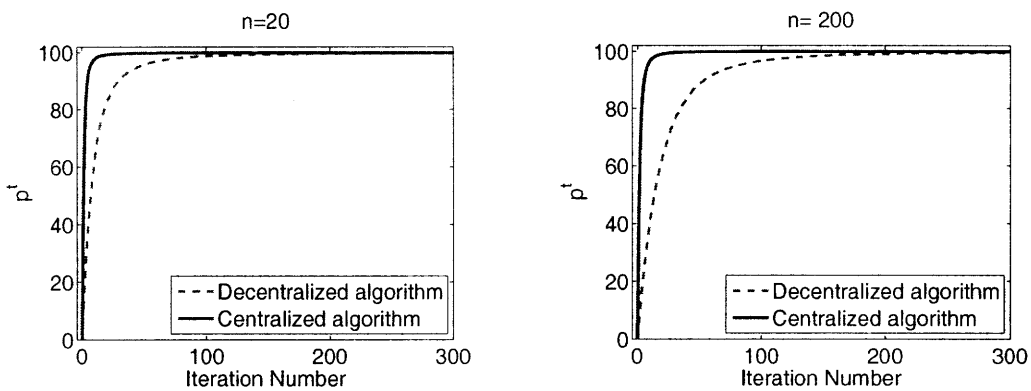

Figure 3-2: Comparison of convergence behavior of the decentralized and centralized algorithms for various n.

solution. Consider the following linear programming problem

$$\begin{aligned} \max_{x_i, i=1,\ldots,n} \quad & \frac{1}{n}\sum_{i=1}^{n} C_i^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} \mathbf{A_i} x_i \leq B \end{aligned} \tag{3.26}$$

where $C_i, x_i \in \Re^q$, $\mathbf{A_i} \in \Re^{m \times q}, i = 1, \ldots, n$, $B \in \Re^m$. It can be seen that (3.26) belongs to the class of problems identified by (3.7). Recall that for a given $\lambda_i \in \Re^m$, $f_i(\lambda_i)$ is the optimal value of the following optimization problem

$$\begin{aligned} \max_{x_i \in \Re^q} \quad & C_i^T x_i \\ \text{s.t.} \quad & \mathbf{A_i} x_i \leq \lambda_i \end{aligned} \tag{3.27}$$

Suppose the dual feasible sets defined by $S_i = \{\nu_i | \mathbf{A_i^T} \nu_i = C_i, \nu_i \geq 0\}$ are non-empty and bounded. It is known from linear programming theory that $f_i(\lambda_i) = \min_{p=1,\ldots,P} \lambda_i^T \nu_{ip}$, where ν_{ip} are the extreme points of the polyhedra defined by S_i. Hence $f_i(\lambda_i)$ is non-differentiable and concave. Further ν_i' is a subgradient of $f_i(\lambda_i)$ at λ_i if and only if it is an optimal solution to the dual problem [9]. Thus if S_i is bounded, it can be seen that Assumption 3.4.3 is satisfied and the convergence analysis of Section 3.4.2 holds.

Let the columns of $\mathbf{A_i}$ be denoted as $\mathbf{a_{ij}}$, $j = 1, \ldots, q$. Also let $C_i = [C_{ij}], j = 1, \ldots, q$. Suppose the column $\mathbf{a_{ik}} > 0$ and $C_{ik} > 0$ for some k such that $1 \leq k \leq q$ and suppose S_i is non-empty. The corresponding dual constraint is $\mathbf{a_{ik}}^T \nu_i = C_{ik}$ showing that S_i is bounded. For the experiments we choose $\mathbf{a_{i1}} = \mathbf{1}$, $i = 1, \ldots, n$ where $\mathbf{1}$ is a vector of ones of the appropriate size. We also choose $C_{i1} = 200$, $i = 1, \ldots, n$. The rest of the constraint matrix and the cost vector are chosen arbitrarily while ensuring that S_i is non-empty.

Although the theoretical results require randomization of the direction of update, it was observed that both the decentralized and the centralized versions of the algorithm converge without the required randomization. Unlike the decentralized algorithm for the differentiable case, there is flexibility in choosing stepsizes. It was observed in the experiments that the practical performance of both the centralized and the decentralized algorithm with or without the randomization of the direction of update depends dramatically on the choice of stepsizes. We present the results of the experiments where the direction of update was not randomized as it provides better insight into convergence behavior of the algorithm. It was observed that convergence was obtained in this case so long as $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$. We choose stepsizes of the form $\gamma^t = \frac{\theta(t)}{2L\sqrt{\log m+1}}\beta^t$, where L is the common Lipschitz constant of the functions f_i, $i = 1, \ldots, n$. Since $C_{i1} = 200$ and $\mathbf{a_{i1}} = \mathbf{1}$ for all i, it can be verified from the dual constraint $\mathbf{a_{i1}}^T \nu_i = C_{i1}$, that $L = C_{i1} = 200$. β^t was chosen to be of the form $\frac{1}{1+w(t)t^{0.51}}$. Thus $\theta(t)$ and $w(t)$ control the rate at which γ^t goes to 0. We chose $w(t)$ as monotonically non-decreasing function bounded above and $\theta(t)$ as a monotonically non-increasing function bounded below. This ensures $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$. For our experiments, we chose $w(0) = 0$ and $w(z\kappa + j) = w(z\kappa)$ for $z = 0, 1, \ldots$ and $j = 1, 2, \ldots, \kappa - 1$ and $w((z+1)\kappa) = min\{w(z\kappa) + r_w, w_{max}\}$. For all the experiments we chose $r_w = 0.0001$ and $w_{max} = 0.1$. We also chose $\theta(t+1) = max\{\theta(t) - r_\theta, \theta_{min}\}$. For these experiments, we chose $\theta(0) = 30$ and $\theta_{min} = 3$ and $r_\theta = 0.1$. We ensured that the union of the communication graphs are connected periodically in the same manner as described in Section 3.6.1. For these experiments, we choose $e_p = 0.5$. Figure 3-3 presents a comparison of the performance of the decentralized algorithm with the centralized algorithm for varying κ. In the figures, n represents the number of agents, q the number of variables per agent and m represents the number of constraints.

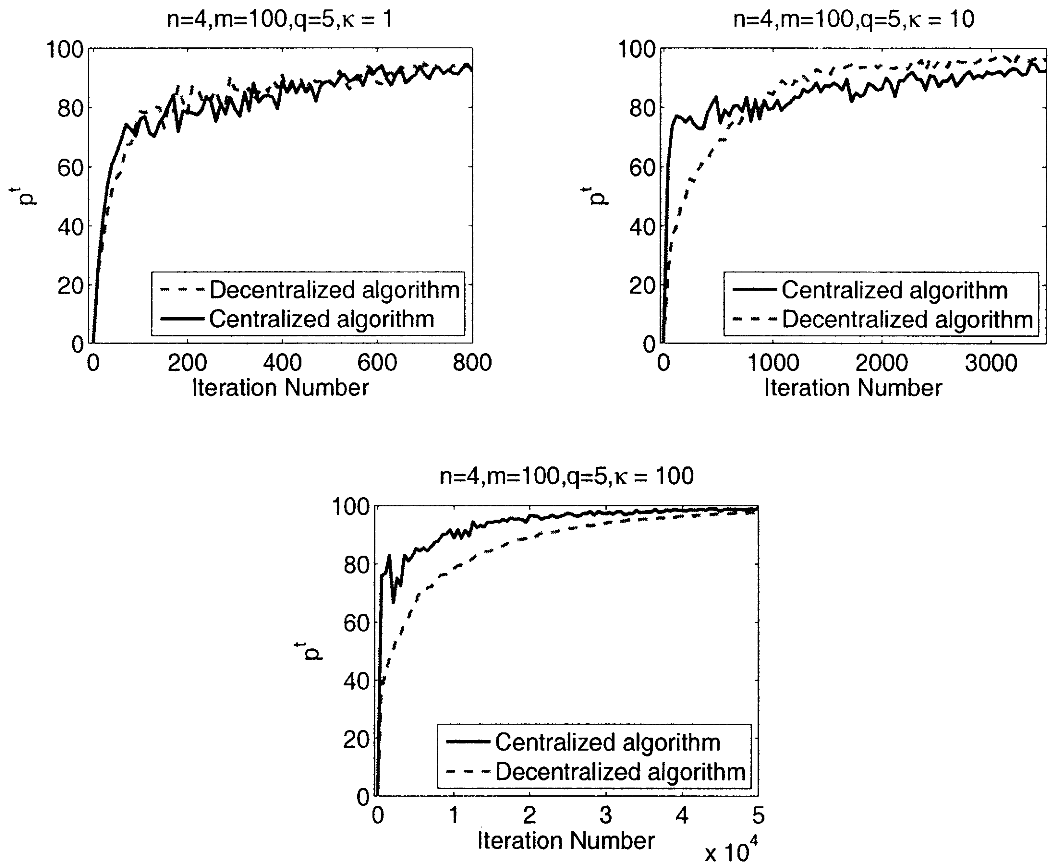

Figure 3-3: Comparison of convergence behavior of the decentralized and centralized algorithms for various κ.

Figure 3-4 presents a comparison of the performance of the decentralized algorithm with the centralized algorithm for varying n. All parameters except $\theta(0)$ were chosen as described previously. $\theta(0)$ was chosen to be 50 for the experiments of Figure 3-4. It can be observed that the performance of the decentralized algorithm scales well with increasing n. The numerical experiments suggest that κ has a greater effect on the practical performance of the algorithm than n.

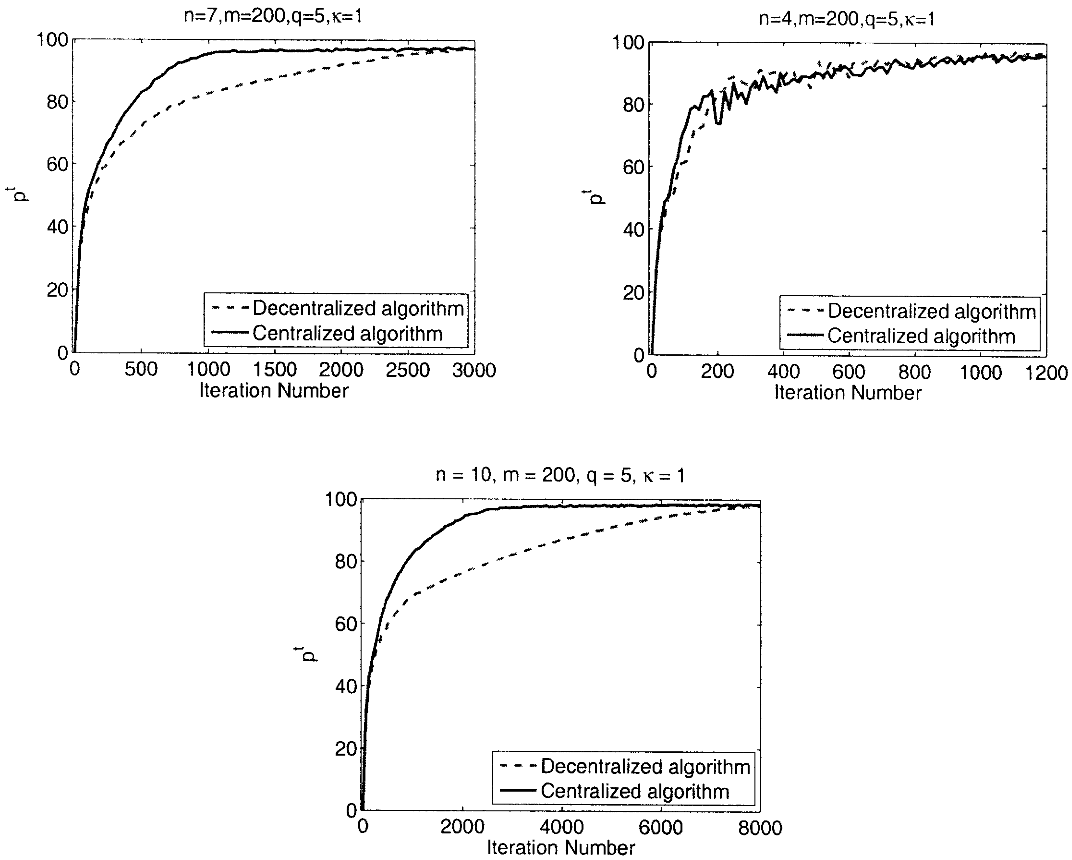

Figure 3-4: Comparison of convergence behavior of the decentralized and centralized algorithms for various n.

Chapter 4

Resource Allocation Problem for Make-To-Order manufacturing firms

4.1 Introduction

The rest of the thesis focuses on a resource allocation problem encountered by firms operating in a Make-To-Order (MTO) manufacturing setting. Such firms do not maintain an inventory of finished products. Production starts only after an order has been placed. MTO manufacturing reduces inventory costs and allows customization of products. In many situations, the firm may be able to realize higher profit by reserving its production time for anticipated future orders with higher profit margins by rejecting an arriving order with lower profit margin. In this work, we view the production time of a MTO manufacturing firm as a resource that needs to be optimally allocated between both currently realized orders as well as future orders. Our focus is to gain insights on the order acceptance policy that a MTO manufacturing firm should adopt.

The current research is motivated by MTO manufacturing firms with fixed production capacity where demand during certain periods may exceed the production capacity. Production time of the firm is thus a scarce resource. A useful control for maximizing the profit of the firm in this case is to dynamically reject orders with unfavorable terms. The relevant terms of the order may include its processing time, reward, lead time. We study the use of order acceptance as a control for maximizing the profit of the MTO manufacturing firm using simple models for the arrival of orders to the firm. The process we envision involves the stochastic arrival of orders with certain terms which the firm has to either accept or reject. We assume in our basic model that the only control available to the firm is to accept or reject an arriving order. In reality, order acceptance is usually an interactive process with the terms of the order being negotiated between the firm and its customer. As pointed out by Gallien *et al.* [24], the analysis of the order acceptance problem would provide useful insights for quoting price, lead time for customers. As an example of how this can be done, we extend the results using our basic model to a problem of quoting lead time and reward to arriving orders. In some cases, rejecting an order may not be an option. For example, it may not be feasible to reject orders from certain customers with whom the firm seeks a long term business relationship. Carr *et al.* [11] accommodate a category of customers whose orders cannot be rejected. It is easy to extend the basic model that we study to accommodate such situations and we show later that some of the results that we establish for the basic model are valid for models that include such special categories of customers. In the next two chapters we study an optimization problem that focuses on selective order acceptance for a firm operating in a purely MTO setting. In this chapter we formulate the order acceptance problem and discuss related work.

4.2 Literature review

The resource allocation problem of order selection in MTO firms can also be viewed as a revenue management problem. Research on revenue management in MTO firms and Make-To-Stock (MTS) firms can be categorized on the basis of the controls used namely price quotation, lead time quotation, order acceptance.

Duenyas *et al.* [21] consider the problem of quoting optimal lead times. They assume that once a lead time is quoted the customer accepts the order with a certain known probability that decreases with increasing quoted lead time. They find a closed form expression for optimal lead time quote for a $GI/GI/\infty$ system and characterize the optimal policy for a $GI/GI/1$ system under the assumption that the scheduling policy for accepted orders is First Come First Served (FCFS).

Duran [22] extends the work of Duenyas *et al.* [21] by considering a model of customer arrival which depends on the past performance of the firm in meeting the deadline. This dependence is in addition to the penalties for missing the deadlines for individual orders. He derives the optimal lead time for a $GI/M/\infty$ case. He shows that the model considered in his work avoids unethical lead time quotes where unachievable lead times are quoted if the revenue from an order is sufficiently large. He also considers the case of a single server and characterizes the optimal policy for this case and analyzes the impact of the modeling assumption that order arrivals are dependent on the past performance of the firm.

Duenyas [20] considers the problem of quoting optimal lead times for a single product facility with multiple customer classes who have different preferences regarding lead times and rewards. He characterizes the optimal solution for lead time quotation when the scheduling policy for accepted orders is FCFS. He shows that the optimal sequencing policy for accepted orders is EDD and proposes a heuristic for quoting lead times based on a solution of the problem with FCFS scheduling policy.

Kapuscinski *et al.* [33] consider a discrete time finite horizon problem of quoting optimal lead times for a production facility with two class of customers who have different lead time preferences under the assumption that all demands are accepted. They characterize the optimal policy and use the characterization to derive heuristics which performs considerably better than other commonly used heuristics in their simulations.

Ray *et al.* [43] consider the problem of finding an optimal static lead time and the capacity for the production facility subject to a constraint on the level of service, when the arrival rate of customers as well as the reward from an order depend on the lead time quoted. Their system consists of only one type of order and all orders are given the same lead time quote. They show that under some conditions the profit from their model can be significantly different from a model which assumes that reward per order is an independent decision variable.

Some of the other works that consider the control of quoting lead time for revenue management are [54], [30], [41].

Carr *et al.* [11] focus on the problem of optimal order acceptance and scheduling for a production facility that makes one type of product to stock (maintains an inventory of the product) while manufacturing another product type to order. They model the production facility as a two product $M/M/1$ queue and characterize the optimal scheduling and order acceptance policy.

The basic order acceptance problem formulated in this chapter is the same as the order acceptance problem studied by Gallien *et al.* [24] who characterize the optimal policy for this problem by showing the existence of a Bellman optimality equation for this problem and an associated differential value function. They also establish an upper bound on the optimal expected reward for this problem in terms of the problem parameters and also show that the policy of accepting all feasible orders is the optimal policy for this problem if the arrival rates for the various order categories are sufficiently low. They develop heuristics for the problem and compare its performance with other commonly used policies. We develop new heuristics for this problem that are computationally inexpensive and are easily extendable to related problems.

Kniker *et al.* [34] focus on an order acceptance problem in which at the most one order arrives at regularly spaced discrete times. The arrival process for orders is Bernoulli and the orders have strict due dates. They show that using a FCFS scheduling policy and solving for an optimal order acceptance policy provides a good improvement in profits over the policy of accepting all feasible orders. DeFregger *et al.* [19] provide heuristics for this problem. We obtain insights on the optimal policy for this problem by characterizing its structure.

Some of the other works that investigate order acceptance for revenue management are [31],

[2]

4.3 Optimization problem statement

The Order Acceptance Problem (OAP) for MTO manufacturing firms formulated in this section is very similar to the problem **(1)** considered by Gallien *et al.* [24].

We model the production facility of the MTO firm as a server that services arriving orders. At any time only one order can be serviced and the other accepted orders wait in an associated queue. We assume that there are three characteristics of an order that are relevant for the decision making process, namely the reward, the processing time and the lead time. The reward for an order is the revenue to the firm from the order minus the production cost and other operating costs associated with completing the order. The processing time for an order is the time required to complete the order if the production facility is completely dedicated for this order. We assume that this processing time is deterministic and is known at the arrival of the order. We define the lead time for an order as the time from the arrival of the order before which it has to be completed if accepted. For example, if the lead time of an order is 5 time units and the order arrives at some time $\bar{t}$, then the order has to be completed by time $\bar{t} + 5$ if accepted. We call the time by which an order is due as its due date or its deadline. We assume that the quoted lead times are reliable in the sense that it is feasible to accept an order only if there exists a schedule for executing the order by its deadline while honoring the deadlines for all previously accepted orders.

We categorize the orders based on their reward, processing time and lead time. We assume that there are a n order categories with $\{\bar{r}_i, \bar{p}_i, \bar{l}_i\}$ denoting the reward, processing time and the lead time for an order belonging to order category $i \in \{1, 2, \ldots, n\}$. The scheduling of accepted orders is preemptive and there are no costs involved with setting up orders or resuming orders. We model the arrival of orders for order category i as a Poisson process with rate $\bar{\lambda}_i$. We assume that the arrival process for the different order categories are independent.

Before we formally describe the optimization problem of interest we wish to comment on the assumptions of the model. The assumption that the processing times are deterministic is not very common in typical queuing systems. However in a manufacturing context, it is reasonable to expect that if the production system is stable the processing time is nearly deterministic. Reliability of the quoted lead time is important in the context of securing business relationships. Since we assume that the processing time of orders is deterministic, it is possible to guarantee the reliability of the due dates by rejecting orders that cannot be completed by their due date.

There are two different decision making problems involved here. Once an order arrives the firm has to decide whether to accept the order or reject it. In the basic model in this section we assume that the firm has the option to reject an arriving order without incurring any penalty. Besides the decision on accepting arriving orders, the firm also has to make decisions regarding scheduling already accepted orders at every point in time. Note that these decisions are inter related. Gallien *et al.* [24] showed that scheduling the accepted orders by their earliest due dates (henceforth denoted as the EDD based scheduling policy) is an optimal scheduling policy. With the assumption of EDD based scheduling policy for accepted orders, decision making (regarding order acceptance) happens only at discrete times and hence they pose a discrete time dynamic programming problem for maximizing the expected average reward per arriving order. In this work, we study the First Come First Served (FCFS) scheduling of accepted orders to construct a computationally feasible approximation to the optimal order acceptance policy. Hence we formulate an optimization problem that allows for a broad class of order scheduling policies such that for every order scheduling policy in this class, a discrete time dynamic programming problem can be formulated for finding the optimal order acceptance policy.

We now formally describe the Order Acceptance Problem. We refer to this problem as OAP-P to highlight the Poisson process assumed for order arrival. At any time t, the state of the queue can be described by $x^q(t) = \{(u_1^t, v_1^t), (u_2^t, v_2^t), \ldots, (u_z(t)^t, v_{z(t)}^t)\}$, where there are $z(t)$ orders in the queue. For convenience we have also included the order being serviced as part of the queue.

For order $k = 1, \ldots, z(t)$, (u_k^t, v_k^t) represents the remaining processing time for the k^{th} order and the time left before the k^{th} order is due. The orders in the queue are numbered in the order of their time of arrival. For example, for integers $k_1 \leq k_2 \leq z(t)$, the order corresponding to k_1 has arrived at an earlier time than the order corresponding to k_2. The state of the system at t can be represented by $x(t) = (x^q(t), j(t))$ where $j(t)$ is the category of the order arriving at time t. In case there is no order arrival at time t, we let $j(t) = 0$. Let $\mathcal{X}^q$ represent the set of all states of the queue and let $\mathcal{X}$ be the set of all the states of the system.

The result of Gallien *et al.* [24] shows the existence of an optimal order scheduling policy (EDD based scheduling policy) that does not depend on the optimal order acceptance policy. We consider stationary order acceptance policies that depend only on the state of the system and stationary order selection policies that depend only on the state of the queue. Let the arrival of an order or completion of an order in the queue be an event. We further require that order scheduling policies that we consider process a fixed order between events. Let $\mathcal{U}$ represent the class of stationary order scheduling policies where a stationary order scheduling policy processes a fixed order between events and uses only the state of the queue for making order scheduling decisions. A policy $u \in \mathcal{U}$ works as follows; If an order arrival event happens, the arriving order is accepted or rejected based on a stationary order acceptance policy which depends only on the state of the system. Once an arriving order is accepted or rejected, an order based on the state of the queue is selected and is processed till another event happens. Similarly, once an order completion event happens, another order is selected based on the state of the queue and is processed till another event happens. We note that the EDD based scheduling policy belongs to this class of scheduling policies. Another policy that belongs to the class $\mathcal{U}$ is the FCFS based scheduling policy. A stationary order scheduling policy $u \in \mathcal{U}$, can be represented as a mapping from the state of the queue to an integer between 1 and the number of orders in the queue. Let $L(x^q)$ be the function that maps a queue state to the number of orders in the queue. For a given state of the queue x^q, let the set $\mathcal{B}(x^q)$ represent the set $\{1, \ldots, L(x^q)\}$ if $L(x^q) > 0$ and let $\mathcal{B}(x^q)$ represent the set $\{0\}$ if $L(x^q) = 0$. The stationary order scheduling policy can be formally defined as $u : \mathcal{X}^q \mapsto \mathcal{B}$. where we use 0 to denote the order scheduling action when there is no order in the queue. All the decisions regarding order acceptance and order scheduling happen only at the events and we now focus on a discrete time optimization problem formulation.

It is clear that given an order scheduling policy $u \in \mathcal{U}$, the feasibility of an arriving order can be determined. For a given state x and an order scheduling policy u, we let $A^u(x)$ denote the feasible action set with respect to order acceptance. $A^u(x) = \{0, 1\}$ if the accepting the arriving order is feasible at state x given the order scheduling policy u and $A^u(x) = \{0\}$ otherwise where 0 represents the action of rejecting the arriving order and 1 represents the action of accepting the arriving order. We let $x_k = (x_k^q, j_k)$ denote the state of the system at the time of the k'th order arrival before the order acceptance decision is taken and let a_k denote the action taken regarding acceptance of the k'th order. A given stationary order scheduling policy $u \in \mathcal{U}$ completely defines the state dynamics. Gallien *et al.* [24] describe the state dynamics for the EDD based scheduling policy for accepted orders.

As another example, we describe the state dynamics if the FCFS based scheduling policy is used for accepted orders. Under the FCFS scheduling policy, the order scheduled for processing at any point of time in the queue is the order that has been in the queue for the longest time. Suppose the state of the queue is $x^q = \{(u_1, v_1), (u_2, v_2), \ldots, (u_z, v_z)\}$ and let the FCFS scheduling policy be applied on the orders in the queue for a time period of τ. If $\tau \geq u_1$, then let $\bar{k}$ be the integer such that $\sum_{i=1}^{\bar{k}} u_i \leq \tau$, otherwise let $\bar{k} = 0$. The state of the queue after τ units of time is $\{(\bar{u}_1, \bar{v}_1), \ldots, (\bar{u}_{\bar{z}}, \bar{v}_{\bar{z}})\}$ where $\bar{z} = z - \bar{k}$ with $\bar{u}_1 = u_{\bar{k}+1} - (\tau - \sum_1^{\bar{k}} u_i)$, $\bar{u}_i = u_{\bar{k}+i}$ for $i = 2, \ldots, z - \bar{k}$ and $\bar{v}_i = v_{\bar{k}+i} - \tau$ for $i = 1, \ldots, z - \bar{k}$. We use the notation of Gallien *et al.* [24] to let $x_k^q \cup j_k$ denote the state of the queue immediately after the k'th order has been accepted. Let $(x^q)(\tau)$ denote the state of the queue starting with state x^q and applying the FCFS scheduling policy for a time period of τ. The state dynamics is as follows

$$
\begin{aligned}
x_{k+1} &= ((x_k^q \cup j_k)(\tau), j_{k+1}), \qquad if \quad a_n = 1 \\
x_{k+1} &= ((x_k^q)(\tau), j_{k+1}), \qquad if \quad a_n = 0
\end{aligned}
$$

where τ is an exponential random variable with mean $\frac{1}{\lambda}$ and j_{k+1} is a discrete random variable such that $P(j_{k+1} = \hat{j}) = \frac{\bar{\lambda}_{\hat{j}}}{\lambda}$ for $\hat{j} = 1, \ldots, n$.

A stationary order scheduling policy $u \in \mathcal{U}$ induces a probability measure

$$\mathcal{P}_u(B|x_k, a_k) = \mathcal{P}_u(x_{k+1} \in B|x_k, a_k)$$

associated with a σ algebra on $\mathcal{X}$, given a state x_k and an order acceptance action a_k. Please see Ritt *et al.* [45] for the relevant mathematical framework. The subscript in the probability measure indicates its dependence on the order scheduling policy u. For a given order scheduling policy, we define a stationary order acceptance policy as $\psi : \mathcal{X} \mapsto A^u$. If the state of the system is x, then under a stationary order acceptance policy ψ, the action taken is $\psi(x)$. For a given order scheduling policy u, an order acceptance policy ψ and an initial state x, we define

$$J(u, \psi, x) = \liminf_{N \to \infty} \frac{E(\sum_{n=0}^{N} g_\psi^u(x_n)|x_0 = x)}{N+1} \tag{4.1}$$

as the expected average reward per order where $g_\psi^u(x_n)$ is the reward for the n'th order under the order scheduling policy u and order acceptance policy ψ. For a given order scheduling policy u, let $\Psi(u)$ denote the class of stationary order acceptance policies. We define

$$J^*(x) = \max_{u \in U} \max_{\psi \in \Psi(u)} J(u, \psi, x) \tag{4.2}$$

as the optimal expected average reward starting from state x. We seek an order scheduling policy

u^* and an order acceptance policy ψ^* such that

$$J^*(x) = J(u^*, \psi^*, x) \tag{4.3}$$

for all states $x \in \mathcal{X}$.

We note that the state space for this problem is not countable. In the next section we define a related discrete time discrete state space problem which can be considered as a both as a different modeling approach for the order acceptance problem as well as an approximation to OAP-P (4.3).

4.4 Discretized order acceptance problem

We now describe an order acceptance problem that uses a different model for order arrival. The set up for this problem is identical to the order acceptance problem described in section (4.3) except that the order arrival process is Bernoulli. In this model orders can arrive only at regularly spaced discrete times indexed by $t = 0, 1, \ldots$. At each time, at the most one order can arrive with probability λ. Given that an order has arrived it belongs to the order category i for $i \in \{1, 2, \ldots, n\}$ with probability $\frac{\lambda_i}{\lambda}$. We denote the reward, the processing time and the lead time for an order category i by r_i, p_i and l_i respectively. We assume that p_i and l_i are integers for all order categories i. We assume that order scheduling decisions are taken only at times $t = 0, 1, \ldots$ and only one order can be scheduled for processing during a time period. Thus this model is discrete time with all the events and decisions happening at times $t = 0, 1, \ldots$. For convenience, we define $\lambda_0 = 1 - \sum_{i=1}^{n} \lambda_i$, the probability that there is no order arrival during a time period.

We let $x_t = \{(u_1^t, v_1^t), (u_2^t, v_2^t), \ldots, (u_{z(t)}^t, v_{z(t)}^t), j^t\}$ represent the state of the system at time t, where there are $z(t)$ orders in the queue. For orders $k = 1, \ldots, z(t)$, (u_k^t, v_k^t) represents the remaining processing time for the k^{th} order and the time left before the k^{th} order is due. j^t is a discrete random variable that represents the category of the order that has arrived at time t. We let

$j^t = 0$ denote the non-arrival of an order at time t. $j^t = i$ with a probability λ_i for $i = 0, 1, \ldots, n$. The random variables j^t are independent and identically distributed. Let $l = \max_{i\in\{1,\ldots,n\}} l_i$. We first observe that the number of orders in the queue at any point of time is finite. To see this, we first note that for $k = 1, \ldots, z(t)$, $u_k^t \geq 1$ since only one order is processed during any one time period and by assumption, the processing times for all order categories are integers. We also note that $\sum_{k=1}^{z(t)} u_k^t \leq max_{k\in\{1,\ldots,z(t)\}} v_k^t \leq l$ because of the assumption that the due dates are reliable. Hence $z(t) \leq l$. It is clear in the light of the above discussion that the state space is finite.

At each time $t = 0, 1 \ldots,$ the decisions to be made at the beginning of the time period are

- If there is an order arrival, then decide whether to accept or reject the order
- Decide the order to be scheduled for processing during the period

We can represent the decision at time t by an ordered pair (a_1^t, a_2^t). We let $a_1^t = 1$ indicate the acceptance of an order and $a_1^t = 0$ represent the rejection of an order. We let a_2^t denote the order that has been scheduled for processing for time t. We let $a_2^t = 0$ in case there is no order scheduled for processing. We note that there is a choice regarding accepting an order only if it is feasible to accept the order while accommodating the deadlines of all accepted orders. Further, there is a choice regarding selecting a specific order for processing during a time period only if it is feasible to complete all accepted orders by their deadlines after processing the selected order for the time period.

We now specify the state transition probabilities for this problem. Let the state of the system at the beginning of time t be $\{(u_1^t, v_1^t), (u_2^t, v_2^t), \ldots, (u_{z(t)}^t, v_{z(t)}^t), j^t\}$ and let $a^t = (0, k)$ be the control applied with $k \in \{1, \ldots, z(t)\}$. That is, the arriving order is rejected and the pending order k is scheduled for processing for the time period. If $u_k^t > 1$, the system transitions to the state $y = \{(u_1^{t+1}, v_1^{t+1}), (u_2^{t+1}, v_2^{t+1}), \ldots, (u_{z(t+1)}^{t+1}, v_{z(t+1)}^{t+1}), j^{t+1}\}$ with probability $\lambda_{j^{t+1}}$ for $j^{t+1} = 0, \ldots, n$ where $z(t+1) = z(t)$, $u_l^{t+1} = u_l^t$ for $l = 1, \ldots, k-1, k+1, \ldots, z(t)$ and $u_k^{t+1} = u_k^t - 1$ and $v_l^{t+1} = v_l^t - 1$ for $l = 1, \ldots, z(t+1)$. In words, the remaining work for all the orders expect the $k'th$ order remain the same while the remaining work for the k^{th} order decreases by 1. The time left before the order is due reduces by 1 for all orders to reflect the passage of a unit of time from t to $t+1$. If $u_k^t = 1$, then the order is complete at time $t+1$ and is removed from the queue and the system transitions to state $\{(u_1^{t+1}, v_1^{t+1}), (u_2^{t+1}, v_2^{t+1}), \dots, (u_{z(t+1)}^{t+1}, v_{z(t+1)}^{t+1}), j^{t+1}\}$ with probability $\lambda_{j^{t+1}}$ for $j^{t+1} = 0, \dots, n$ where $z(t+1) = z(t) - 1$, $u_l^{t+1} = u_l^t$ for $l = 1, \dots, k-1$ and $u_l^{t+1} = u_{l+1}^t$ for $l = k, \dots, z(t+1)$ and $v_l^{t+1} = v_l^t - 1$ for $l = 1, \dots, k-1$ and $v_l^{t+1} = v_{l+1}^t - 1$ for $l = k, \dots, z(t+1)$. If the arriving order is accepted and scheduled for processing and if $p_{j^t} > 1$ then the system transitions to the state $\{(u_1^{t+1}, v_1^{t+1}), (u_2^{t+1}, v_2^{t+1}), \dots, (u_{z(t+1)}^{t+1}, v_{z(t+1)}^{t+1}), j^{t+1}\}$ with probability $\lambda_{j^{t+1}}$ for $j^{t+1} = 0, \dots, n$ where $z(t+1) = z(t) + 1$, $u_l^{t+1} = u_l^t$ for $l = 1, \dots, z(t+1) - 1$ and $u_{z(t+1)}^{t+1} = p_{j^t} - 1$ and $v_l^{t+1} = v_l^t - 1$ for $l = 1, \dots, z(t+1) - 1$ and $v_{z(t+1)}^{t+1} = l_{j^t} - 1$. If the arriving order is accepted and scheduled for processing and if $p_{j^t} = 1$, then the order is completed during the time period and the system transitions to the state $\{(u_1^{t+1}, v_1^{t+1}), (u_2^{t+1}, v_2^{t+1}), \dots, (u_{z(t+1)}^{t+1}, v_{z(t+1)}^{t+1}), j^{t+1}\}$ with probability $\lambda_{j^{t+1}}$ for $j^{t+1} = 0, \dots, n$ where $z(t+1) = z(t)$, $u_l^{t+1} = u_l^t$ for $l = 1, \dots, z(t+1)$ and $v_l^{t+1} = v_l^t - 1$ for $l = 1, \dots, z(t+1)$. The state transition probabilities for the case when the arriving order is accepted and an order k that is different from the arriving order is scheduled for processing can be given in a similar fashion.

We model the problem of maximizing the expected average reward per time as an infinite horizon average cost MDP. Note that we consider scheduling and order acceptance jointly in this problem. For a stationary order scheduling and order acceptance policy μ, let J_μ be defined as follows.

$$J_\mu(x) = \liminf_{T \to \infty} \frac{1}{T} E \left[\sum_{t=1}^{T} g_\mu(x_t) | x_0 = x \right] \tag{4.4}$$

where $g_\mu(x_t)$ is the reward obtained at time t by following the policy μ and x_t is the state of the system at time t.

The optimization problem of interest is to find a policy μ^* such that

$$J_{\mu^*}(x) = max_{\mu} J_{\mu}(x) \tag{4.5}$$

for all states x.

We refer to the order acceptance problem described in this section as OAP-B to highlight the Bernoulli arrival process used for modeling order arrivals. In this work we view OAP-B as an approximation to OAP-P with an appropriate choice of the parameter values. In particular we study OAP-B with the idea of constructing approximations to the optimal policy for OAP-P. Finding the optimal policy for OAP-B is computationally challenging due to the size of the state space. In the next chapter we assume that the FCFS based scheduling policy is used for accepted orders which considerably reduces the size of the state space. We present results for OAP-B under this assumption.

Chapter 5

Order Acceptance Problem (Bernoulli) with First Come First Served Scheduling policy

In this chapter we study OAP-B (4.5) in detail under the assumption of FCFS based scheduling of accepted orders. The use of FCFS based scheduling policy for accepted orders ensures fair access to the production facility for all the customers. In some cases it is a common practice to cite a common lead time for all orders [21]. In case all orders are quoted a common lead time, the FCFS scheduling policy for accepted orders is the same as EDD based scheduling of accepted orders. The reward for a FCFS scheduling policy does not changes even if there are costs associated with interrupting and resuming orders. Also, while this assumption renders the resulting policy sub-optimal for OAP-B (4.5) except under special circumstances, it greatly reduces the state space of the underlying problem.

In this chapter we establish some structural results for the OAP-B under FCFS scheduling of accepted orders. We also investigate a special class of policies called the static policies that are easy to implement from a practical point of view. A quantity of interest that will play an important role in our analysis is the time needed to finish already accepted but uncompleted orders which we refer to as the remaining work. We recall from Chapter 4 that the state of the system at time t for OAP-B can be represented by $x_t = \{(u_1^t, v_1^t), (u_2^t, v_2^t), \ldots, (u_{z(t)}^t, v_{z(t)}^t), j^t\}$ where there are $z(t)$ orders in the queue at time t. For order $k = 1, \ldots, z(t)$, (u_k^t, v_k^t) represents the remaining processing time for the $k'th$ order and the time left before the $k'th$ order is due. j^t is a discrete random variable that represents the category of the order that has arrived at time t. We define the remaining work at this time as $\sum_{k=1}^{z(t)} u_k^t$. We establish later in this Chapter that under the assumption of FCFS scheduling of accepted orders, the feasibility of an order is completely determined by the remaining work at that time. We also establish in this chapter that in general there exists an optimal order acceptance policy that rejects an order if the remaining work at the time of order arrival is above a threshold that depends on the order category. We start our analysis by considering a special case of the problem OAP-B (4.5) that admits an easy computation of the optimal policy.

5.1 Special case; No waiting room in the queue

Recall that we model the production facility of the firm as a queue to which orders arrive via a Bernoulli process. We assume that the orders can be divided into n categories based on their reward, processing time and lead time. In the following, let C be the set $\{1, 2, \ldots, n\}$ and let $\bar{C}$ be the set $\{0, 1, 2, \ldots, n\}$. We let r_i, p_i, l_i denote the reward, processing time and lead time for order category i and we let λ_i denote the arrival rate of order category i for $i \in C$. We define $\lambda_0 = 1 - \sum_{i=1}^n \lambda_i$. At each time $t = 0, 1, \ldots,$ at the most one order can arrive with probability $\lambda = \sum_{i=1}^n \lambda_i$. Given that an order has arrived it belongs to the order category i for $i \in \{1, 2, \ldots, n\}$ with probability $\frac{\lambda_i}{\lambda}$. In this section, we consider the order acceptance problem when there is no waiting room in the queue and the processing of an accepted order has to begin immediately and must be processed to completion before another order could be accepted. In other words $l_i = p_i$ for all order categories $i \in C$. Thus FCFS is the only scheduling policy for this problem and we need to compute only the optimal order acceptance policy for this problem.

We denote by w^t, the remaining work at time t. If no order is being processed at time t, then

$w^t = 0$. Let j^t denote the category of an arriving order at time t. We let $j^t = 0$ if there is no order arrival at time t. For simplicity we also let $j^t = 0$ when $w^t > 0$, since the arriving order would be rejected regardless of its category since there is no waiting room in the queue. The state of the system at time t for this problem can be modeled by $x_t = (w^t, j^t, i^t)$ where i^t denotes the type of the order being processed at time t. If no order is being processed, then $i^t = 0$. Storing the type of the order being processed as part of the state information is not necessary. However it simplifies the computation of the associated steady state probabilities as we will see. We let $\mathcal{S}$ denote the state space for this problem. The system has a choice of accepting an arriving order only when no order is being processed. Thus the system has a choice regarding accepting or rejecting an arriving order only for states $(0, j, 0)$ where $j \in C$. We let $A(x)$ denote the set of actions available when the state of the system is x. $A(x) = \{0, 1\}$ if accepting the arriving order is feasible at state x and $A(x) = \{0\}$ otherwise where 0 represents the action of rejecting the arriving order and 1 represents the action of accepting the arriving order.

We now describe the state transition structure for this problem. Consider the state $x_t = (0, \bar{j}, 0)$. If the control action is to accept the order then the system deterministically transitions to the state $x_{t+1} = (p_{\bar{j}} - 1, 0, \bar{j})$. This is because, the order arrives at time t and at time $t+1$ the order has already been processed for a unit of time and so the remaining work at time $t+1$, $w^{t+1} = p_{\bar{j}} - 1$. If we reject the order at state $x_t = (0, \bar{j}, 0)$, then the system transitions to the state $(0, j, 0)$ with a probability λ_j for $j \in \bar{C}$. If the system is in the state $(m, 0, i)$ it deterministically transitions to the state $(m-1, 0, i)$ for $p_i - 1 \geq m > 1$. This is because for $p_i - 1 \geq m > 1$ all arriving orders are rejected and by our notational choice, $j^t = 0$ when $w^t > 0$. From the state $(1, 0, i)$, the system transitions to the state $(0, j, 0)$ with a probability λ_j for $j \in \bar{C}$.

5.1.1 Stationary policy description

A stationary policy is a function that associates with each state of the system a feasible action. From the problem definition, it is clear that at time t, there is a choice regarding order acceptance only when there is an order arrival at that time and $w^t = 0$. Thus we can describe every stationary policy μ for this problem by a set of order categories that the policy accepts on arrival when no order is being processed. For example, the policy that accepts all feasible orders can be described by the set $\{1, 2, \ldots, n\}$.

5.1.2 Problem Formulation

For the stationary policy μ, we define the expected average reward per time $J_\mu(x)$ starting from state x as follows

$$J_\mu(x) = \liminf_{T\to\infty} \frac{1}{T} E\left[\sum_{t=1}^{T} g_\mu(x_t)|x_0 = x\right] \tag{5.1}$$

where $g_\mu(x_t)$ is the reward obtained at time t by following the policy μ and x_t is the state of the system at time t.

The optimization problem of interest is to find a policy μ^* such that

$$J_{\mu^*}(x) = max_\mu J_\mu(x) \tag{5.2}$$

for all states x.

5.1.3 Characterization of the optimal policy

A recurrent class for a finite state Markov chain is a set of states such that for any pair of states in the set, the probability of eventually reaching one state from the other is one (see Bertsekas and Tsitsiklis [8]). Every Markov chain has at least one recurrent class [8]. Let $\mathcal{P}(x_t = \bar{x}|x_0 = \bar{y})$ denote the probability that for a stationary Markov chain, $x_t = \bar{x}$ given that $x_0 = \bar{y}$. We first formalize an observation regarding general finite state stationary Markov chains

Proposition 5.1.1. *Consider a finite state stationary Markov chain. If there exists a state $\bar{x}$ such that for every state y there exists some t_y with $\mathcal{P}(x_{t_y} = \bar{x}|x_0 = y) > 0$, then the Markov chain has* *a single recurrent class.*

Proof. We know that every finite state Markov Chain has at least one recurrent class [8]. Consider a recurrent class $\mathcal{M}$. By the assumption in the proposition, for any state $y \in \mathcal{M}$, there exists some t_y with $\mathcal{P}(x_{t_y} = \bar{x}|x_0 = y) > 0$. Hence $\bar{x}$ belongs to the recurrent class $\mathcal{M}$. Since this is true for all recurrent classes, $\bar{x}$ belongs to all recurrent classes showing that the Markov chain has a single recurrent class.

Every stationary policy for a MDP induces a stationary Markov chain. A stationary policy is unichain if the Markov chain induced by it has only one recurrent class [5]. We claim that every stationary policy for the OAP-B (5.2) with no waiting room in the queue is unichain. To see this, consider a stationary policy μ. As explained before, μ is described by the set of order categories that are accepted. Let $\mathcal{P}_\mu(x_t = \bar{x}|x_0 = \bar{y})$ be the probability that $x_t = \bar{x}$ given that $x_0 = \bar{y}$ under the policy μ. For any stationary policy μ, it can be verified that $i \in \mathcal{C}$, $\mathcal{P}_\mu(x_m = (0,0,0)|x_0 = (m,0,i)) = \lambda_0$ for $p_i - 1 \geq m \geq 1$. Also, for $j \in \mu$, $\mathcal{P}_\mu(x_{p_j} = (0,0,0)|x_0 = (0,j,0)) = \lambda_0$ and for $j \notin \mu$, $\mathcal{P}_\mu(x_1 = (0,0,0)|x_0 = (0,j,0)) = \lambda_0$. Hence the state $(0,0,0)$ satisfies the assumption of proposition 5.1.1 for the Markov chain induced by the stationary policy μ and hence has a single recurrence class.

Thus condition (1) of proposition 4.2.6 of [5] is satisfied and therefore the optimal expected average reward per time is the same starting from all states. Also, there exists a solution to the Bellman's equation for this problem. Let J^* denote the common optimal expected average reward per time. J^* satisfies the following Bellman equation for this problem:

$$
\begin{aligned}
J^* + h(0,0,0) &= \sum_{j=0}^{n} \lambda_j h(0,j,0), && && (5.3)\\
J^* + h(0,i,0) &= max\{r_i + h(p_i - 1, 0, i), \sum_{j=0}^{n} \lambda_j h(0,j,0)\}, && i \in \mathcal{C} && (5.4)\\
J^* + h(m,0,i) &= h(m-1,0,i), \qquad p_i \geq m > 1 && i \in \mathcal{C} && (5.5)\\
J^* + h(1,0,i) &= \sum_{j=0}^{n} \lambda_j h(0,j,0), \qquad i \in \mathcal{C} && && (5.6)
\end{aligned}
$$

where the vector h is an associated optimal differential value function. Consider some $i \in \mathcal{C}$. From equations (5.5) and (5.6), it can be seen that $h(p_i - 1, 0, i) = \sum_{j=0}^{n} \lambda_j h(0,j,0) - (p_i - 1)J^*$. Hence,

$$
\begin{aligned}
max\{r_i + h(p_i - 1, 0, i), \sum_{j=0}^{n} \lambda_j h(0,j,0)\} &= max\{r_i + \sum_{j=0}^{n} \lambda_j h(0,j,0) - (p_i - 1)J^*, \sum_{j=0}^{n} \lambda_j h(0,j,0)\}\\
&= max\{r_i - (p_i - 1)J^*, 0\}
\end{aligned}
$$

If $r_j \geq (p_j - 1)J^*$, then accepting the order is an optimal control action at state $(0,j,0)$. Therefore the policy μ^* defined by the set $\{j : \frac{r_j}{p_j - 1} \geq J^*; j \in \mathcal{C}\}$ is an optimal stationary policy. We have shown that there exists an optimal stationary policy that has a threshold structure, that is, it accepts all feasible orders with a ratio of $\frac{r_i}{p_i - 1}$ above a certain threshold. We now provide an expression for the expected average reward per time of any stationary policy which when used together with this structural result provides a simple algorithm for finding an optimal stationary policy for this problem.

5.1.4 Derivation of steady state probabilities

A Markov chain with a single recurrent class is said to be periodic if the states of the recurrent class can be grouped into k disjoint subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ with $k > 1$ such that if the system is in a state $\bar{x} \in S_i$, it can only transition to states in the set S_{i+1} for $1 \leq i \leq k-1$ and if the system is in a state

$\bar{x} \in S_k$, it can only transition to states in the set S_1. A Markov chain with a single recurrent class that is not periodic is said to be aperiodic. We observe that a Markov chain with a single recurrent class consisting of at least one state with a non-zero one step transition probability from that state to itself cannot be periodic. For any stationary policy μ, the associated recurrence class $\mathcal{S}_\mu$ consists of the state $(0,0,0)$ as shown in Section (5.1.3). Also, $\mathcal{P}_\mu(x_1 = (0,0,0)|x_0 = (0,0,0)) = \lambda_0 > 0$ and hence the Markov chain associated with any stationary policy μ is aperiodic. It is known that an aperiodic Markov chain with a single recurrent class has a unique set of associated steady state probabilities [8]. We now derive the steady state probabilities for this problem and use them to find an expression for the expected average reward per time for a stationary policy. Let $\theta_\mu(x)$ denote the steady state probability of the state x under the stationary policy μ. The balance equation for the steady state probability for the state $(p_i, 0, i)$ for some $i \in \mu$ is,

$$\theta_\mu(p_i - 1, 0, i) = \theta_\mu(0, i, 0) \tag{5.7}$$

For $p_i - 1 \geq m > 1$,

$$\theta_\mu(m - 1, 0, i) = \theta_\mu(m, 0, i) \tag{5.8}$$

From equations (5.7) and (5.8), we conclude that that for $i \in \mu$ and $m = p_i - 1, \ldots, 1$

$$\theta_\mu(m, 0, i) = \theta_\mu(0, i, 0) \tag{5.9}$$

Also, for all $i \in \bar{\mathcal{C}}$,

$$\begin{aligned} \theta_\mu(0, i, 0) &= \sum_{j \in \mu} \lambda_i \theta_\mu(1, 0, j) + \sum_{j \notin \mu} \lambda_i \theta_\mu(0, j, 0) \\ &= \lambda_i \sum_j \theta_\mu(0, j, 0) \end{aligned} \tag{5.10}$$

The last equality follows from (5.9). Equations (5.9), (5.10) and the normalizing equation that sum of the steady state probabilities over all the states must be 1 provide a set of linear equations that define the steady state probabilities. It can be verified that for all i,

$$\theta_\mu(0, i, 0) = \frac{\lambda_i}{1 - \sum_{j\in\mu} \lambda_j + \sum_{j\in\mu} \lambda_j p_j}$$

As noted before, the expected average reward per time starting at state x, defined by $J_\mu(x) = \lim_{T\to\infty} \frac{1}{T} E\left[\sum_{t=1}^{T} g_\mu(x_t)|x_0 = x\right]$ exists and is the same for all states x. Let J_μ denote the common expected average reward per time obtained by following the policy μ. Using the steady state probabilities derived for this problem, J_μ can be expressed by the following equation.

$$J_\mu = \frac{\sum_{i\in\mu} \lambda_i r_i}{1 - \sum_{j\in\mu} \lambda_j + \sum_{j\in\mu} \lambda_j p_j} \tag{5.11}$$

5.1.5 A simple algorithm for finding Optimal policy

We assume, without loss of generality that $\frac{r_1}{p_1 - 1} \geq \frac{r_2}{p_2 - 1} \geq \cdots \geq \frac{r_n}{p_n - 1}$. From the characterization of the optimal policy in section (5.1.3), there exists some i^* such that the stationary policy described by the set $\{1, \ldots, i^*\}$ is an optimal policy. Let the expected average reward per time for all initial states of the stationary policy defined by the set $\{1, \ldots, i\}$ be J^i. From equation (5.11), it can be seen that $J^i = \frac{\sum_{j=1}^{i} \lambda_j r_j}{1 - \sum_{j=1}^{i} \lambda_j + \sum_{j=1}^{i} \lambda_j p_j}$. Thus the optimal expected average reward per time $J^* = \max_{i=1,\ldots,n} J^i$ and the stationary policy $\{1, \ldots, i^*\}$ is optimal where $i^* = \underset{i=1,\ldots,n}{\operatorname{argmax}} J^i$

5.2 Analysis of OAP-B with FCFS scheduling policy

The assumption that FCFS is the scheduling policy for accepted orders hugely reduces the state space, as it is only necessary to keep track of remaining work at any time. It is not necessary to store the exact arrival times and the due dates of the orders in the queue. In this section we formulate the problem OAP-B under the assumption that FCFS is the scheduling policy for accepted orders. The formulation is identical to the problem formulated by Kniker *et al.* [34]. Using numerical simulations, they show the benefits of using the optimal policy for this problem over the policy of accepting all feasible orders. However they do not provide any characterization of the optimal policy for this problem. Similar to our approach in (5.1), we show that there exists a solution to the Bellman equation for this problem and gain some insights on the structure of the optimal order acceptance policy through an analysis of the Bellman equation.

5.2.1 State space and state transition structure

We recall the set up for OAP-B (4.5). We assume that the orders can be grouped into n categories based on reward, lead time and processing time and we let r_i, p_i, l_i denote the reward, processing time and lead time for order category i. We assume that at the most one order can arrive at any time with a probability $\lambda = \sum_{i=1}^{n} \lambda_i$ and given that an order has arrived, it belongs to order category i with probability $\frac{\lambda_i}{\lambda}$. Let w^t be the remaining work at time t. Let j^t denote the type of the arriving order at time t. We let $j^t = 0$ to denote the non-arrival of an order at time t. As before, let C be the set $\{1, 2, \ldots, n\}$ and $\bar{C}$ be the set $\{0, 1, \ldots, n\}$ and let $\lambda_0 = 1 - \sum_{j=1}^{n} \lambda_j$. For convenience, we make the assumption that $p_i > 2$ for $i \in C$ and that $\lambda_0 > 0$. This does not represent any loss of generality for the purpose of using an appropriately defined problem of type OAP-B for approximating a given problem of type OAP-P as we will see in Chapter 6.

We claim that the feasibility of an arriving order at time t can be decided with the knowledge of w^t. To see this note that since the scheduling policy for accepted orders is FCFS, an arriving order belonging to category $i \in C$ at time t would be scheduled for processing between times $t + w^t$ and $t + w^t + p_i$ if accepted. Hence it is feasible to accept an order arriving at time t if and only if $w^t + p_i \leq l_i$. Under the FCFS scheduling policy for accepted orders, we model the state of the system at time t as $x_t = (w^t, j^t)$. For convenience, we define $l = \max_{i \in C} l_i$. For those states where it is feasible to accept an arriving order, there is a choice regarding accepting or rejecting the order.

For all other states the arriving order is rejected. We let S be the set of all the states that the system can be in.

We now describe the state transition structure. Suppose the state of the system is $(w, i), i \in \mathcal{C}$ and it is feasible to accept the order. If the order is accepted, the system transitions to the state $(w + p_i - 1, j)$ with a probability λ_j for $j \in \bar{\mathcal{C}}$. Suppose the state of the system is (w, i) , $i \in \bar{\mathcal{C}}$. If the order is rejected then the system transitions to the state $(w - 1, j)$ with a probability λ_j if $w > 0$. If $w = 0$ then the system transitions to the state $(0, j)$ with a probability λ_j for $j \in \bar{\mathcal{C}}$.

5.2.2 Problem Formulation

Let $A(x)$ denote the feasible action set with respect to order acceptance. $A(x) = \{0, 1\}$ if the accepting the arriving order is feasible at state x and $A(x) = \{0\}$ otherwise where 0 represents the action of rejecting the arriving order and 1 represents the action of accepting the arriving order. We define a stationary order acceptance policy as a mapping from the set of states to the set of feasible actions, $\mu : S \mapsto A$. For a given stationary policy μ, we define

$$J_\mu(x) = \liminf_{T\to\infty} \frac{1}{T} E\left[\sum_{t=1}^{T} g_\mu(x_t)|x_0 = x\right] \tag{5.12}$$

where $g_\mu(x_t)$ is the reward obtained at time t by following the policy μ and x_t is the state of the system at time t.

The optimization problem of interest is to find a policy μ^* such that

$$J_{\mu^*}(x) = max_\mu J_\mu(x) \tag{5.13}$$

for all states x.

5.2.3 Characterization of Optimal Policy

Consider a stationary policy μ. We first note that $\mathcal{P}_\mu(x_w = (0,0)|x_0 = (w,0)) \geq (\lambda_0)^w > 0$ for $l-1 \geq w \geq 0$. We also note that for any state $x = (w,i)$ such that $\mu(x) = 1$, $\mathcal{P}_\mu(x_1 = (w+p_i-1,0)|x_0 = (w,i)) = \lambda_0 > 0$. Therefore, for any state $x = (w,i)$ such that $\mu(x) = 1$, $\mathcal{P}_\mu(x_{w+p_i} = (0,0)|x_0 = (w,i)) > 0$. For a state $x = (w,i)$ with $w > 0$, such that $\mu(x) = 0$, $\mathcal{P}_\mu(x_1 = (w-1,0)|x_0 = (w,i)) = \lambda_0 > 0$ and hence $\mathcal{P}_\mu(x_w = (0,0)|x_0 = (w,i)) > 0$. Finally, we note that for a state $x = (0,i)$ such that $\mu(x) = 0$, $\mathcal{P}_\mu(x_1 = (0,0)|x_0 = (0,i)) = \lambda_0 > 0$. Thus the state $(0,0)$ satisfies the assumption of Proposition 5.1.1 and we conclude that the Markov chain induced by any stationary policy has a single recurrence class. Thus condition (1) of proposition 4.2.6 of [5] is satisfied and the optimal expected average reward per time is the same starting from all states. Also there exists a solution to the Bellman's equation for this problem. Since, $\mathcal{P}_\mu(x_1 = (0,0)|x_0 = (0,0)) = \lambda_0 > 0$ and the state $(0,0)$ belongs to the only recurrence class, we note that the Markov chain induced by any stationary policy is also aperiodic. Let J^* denote the common optimal expected average reward per time. We now establish some properties of the optimal policy.

Lemma 5.2.1. *Consider an order category $\bar{i}$ with reward $r_{\bar{i}}$ and processing time $p_{\bar{i}}$. If $\frac{r_{\bar{i}}}{p_{\bar{i}}} \geq J^*$, then it is optimal to accept an order of category $\bar{i}$ whenever feasible.*

Proof. Consider some state (w,i) where $w \geq 1$. If an order of category i is feasible at state (w,i), the optimality equation for the state (w,i) can be written as

$$J^* + h(w,i) = \max\{r_i + \sum_{j=0}^{n} \lambda_j h(w+p_i-1,j), \sum_{j=0}^{n} \lambda_j h(w-1,j)\} \tag{5.14}$$

where h is an optimal differential value function that satisfies the Bellman equation. Otherwise the optimality equation for the state (w, i) is given by

$$J^* + h(w, i) = \sum_{j=0}^{n} \lambda_j h(w-1, j) \tag{5.15}$$

For convenience, we define $\bar{h}(w) = \sum_{j=0}^{n} \lambda_j h(w, j)$ for all w. From (5.14) and (5.15), we have that,

$$J^* + h(w, i) \geq \bar{h}(w-1) \qquad \forall w \geq 1, \quad \forall i \in \bar{C} \tag{5.16}$$

Multiplying by λ_i on both sides and summing over $i \in \bar{C}$,

$$\sum_{i=0}^{n} \lambda_i (J^* + h(w, i)) \geq \sum_{i=0}^{n} \lambda_i \bar{h}(w-1)$$

Since $\sum_{i=0}^{n} \lambda_i = 1$, it follows that

$$J^* + \bar{h}(w) \geq \bar{h}(w-1) \tag{5.17}$$

Consider some state $(w, \bar{i})$ with $w \geq 1$ where accepting an order of category $\bar{i}$ is feasible. By repeated application of (5.17), it can be seen that

$$\bar{h}(w + p_{\bar{i}} - 1) \geq \bar{h}(w-1) - p_{\bar{i}} J^*$$

If $r_{\bar{i}} \geq p_{\bar{i}} J^*$, then from the above equation $r_{\bar{i}} + \bar{h}(w + p_{\bar{i}} - 1) \geq \bar{h}(w-1)$ and from (5.14) it is clear that it is optimal to accept an order of type $\bar{i}$ whenever $w \geq 1$ and the order is feasible.

When $w = 0$, the optimality equation for a feasible order of type $\bar{i}$ is given by

$$J^* + h(0, \bar{i}) = \max\{r_{\bar{i}} + \bar{h}(p_{\bar{i}} - 1), \bar{h}(0)\}$$

By repeated application of (5.17), it can be seen that

$$\bar{h}(p_{\bar{i}} - 1) \geq \bar{h}(0) - (p_{\bar{i}} - 1)J^*$$

Hence, if $r_{\bar{i}} \geq p_{\bar{i}}J^* \geq (p_{\bar{i}} - 1)J^*$, it is optimal to accept an order of type $\bar{i}$ at the state $(0, \bar{i})$. Thus, if $r_{\bar{i}} \geq p_{\bar{i}}J^*$, it is optimal to accept an order of type $\bar{i}$ whenever it is feasible. □

Lemma (5.2.1) suggests that when the rewards grow proportionately with the processing time, then for the OAP-B with FCFS as the scheduling policy for accepted orders, it is optimal to accept all feasible orders. We establish this result in the following proposition

Proposition 5.2.1. *Suppose $r_i = bp_i$ for $i \in C$ where b is some constant. Then it is optimal to accept all feasible orders*

Proof. Let J^* be the optimal expected average reward per time starting from all initial states. Consider a situation where instead of being paid immediately, a reward of $\frac{r_i}{p_i}$ is paid per unit time while an order of category i is processed. The expected average reward per time under any policy with this scheme of payment is the same as the system where all the reward is paid as soon as an order is accepted. Hence, $J^* \leq \max_{i=1,2,\ldots,n}\{\frac{r_i}{p_i}\}$. Therefore under the assumptions of the proposition, $J^* \leq b$. The proposition follows from Lemma 5.2.1 □

We now establish one of the main results of this Chapter. The result provides insight into the structure of the optimal policy.

Theorem 5.2.1.

Suppose it is optimal to accept an order of category $\bar{i}$ at state $(\bar{w}, \bar{i})$. Then it is also optimal to accept an order of type $\bar{i}$ at all states $(w, \bar{i})$ with $w < \bar{w}$.

Proof. The proof for this theorem is based on the following lemma

Lemma 5.2.2. *Let h be an optimal differential value function for the problem (5.13). For $w = 0, 1, \ldots, l-1$, let $\bar{h}(w) = \sum_{j=0}^{n} \lambda_j h(w, j)$.*

1. *For $0 < w < l-1$, $\bar{h}(w) - \bar{h}(w+1) \geq \bar{h}(w-1) - \bar{h}(w)$.*

2. *For $0 < w \leq l-1$, $\bar{h}(w-1) - \bar{h}(w) \geq 0$*

Proof. Recall that $l = \max_{i \in 1,2,\ldots,n} l_i$. Let the order categories be numbered such that $p_1 \leq \cdots \leq p_n$. An arriving order belonging to category i is feasible if and only if the remaining work w at the time of order arrival satisfies $w + p_i \leq l_i$. Therefore any order arriving with the remaining processing time w such that $l - p_1 + 1 \leq w \leq l-1$ is infeasible. Note that the maximum value of remaining work at the beginning of any time period is $l-1$. Thus for $l - p_1 + 1 \leq w \leq l-1$ and $i \in \bar{\mathcal{C}}$ the Bellman's equation is

$$J^* + h(w, i) = \bar{h}(w-1) \tag{5.18}$$

For w, such that $l - p_1 + 1 \leq w \leq l-1$, multiplying the above equation by λ_i and summing over $i \in \bar{\mathcal{C}}$, we get $\bar{h}(w-1) = \bar{h}(w) + J^*$. Since $p_1 > 2$ by assumption, $\bar{h}(l-2) - \bar{h}(l-1) = \bar{h}(l-3) - \bar{h}(l-2)$ and thus the lemma is true for $w = l-2$. We prove the lemma by induction. Assume that $\bar{h}(w) - \bar{h}(w+1) \geq \bar{h}(w-1) - \bar{h}(w)$ for $l-1 > w \geq \tilde{w}+1$ where $\tilde{w} > 0$. Under this assumption, we show $\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1) \geq \bar{h}(\tilde{w}-1) - \bar{h}(\tilde{w})$. This completes the proof of the first part of the lemma.

For $0 \leq w \leq l-2$, let $\mathcal{J}_w$ represent the set of order categories such that it is optimal to accept an order of category $j \in \mathcal{J}_w$ at states $(w+1, j)$ and (w, j). Let $\mathcal{I}_w$ represent the set of order categories such that it is optimal to reject an order of category $i \in \mathcal{I}_w$ at states $(w+1, i)$ and (w, i). Let the set $\mathcal{K}_w$ represent the set of order categories such that it is optimal to accept an order of category $k \in \mathcal{K}_w$ at state $(w+1, k)$ and it is optimal to reject an order of category $k \in \mathcal{K}_w$ at state (w, k). Let $\mathcal{M}_w$ represent the set of order categories such that it is optimal to accept an order of category $m \in \mathcal{M}_w$ at state (w, m) but is optimal to reject an order of category $m \in \mathcal{M}_w$ at state $(w+1, m)$. Let $\lambda_{\mathcal{I}_w} = \sum_{i \in \mathcal{I}_w} \lambda_i$ and let $\lambda_{\mathcal{J}_w}, \lambda_{\mathcal{K}_w}, \lambda_{\mathcal{M}_w}$ be defined similarly. $\lambda_{\mathcal{I}_w} > 0$ since the order category $0 \in \mathcal{I}_w$ for all w and $\lambda_0 > 0$. From the optimality equations (5.14) and (5.15) for an order category $j \in \mathcal{J}_{\tilde{w}} \cup \mathcal{K}_{\tilde{w}}$, we have

$$J^* + h(\tilde{w}+1, j) = r_j + \bar{h}(\tilde{w} + p_j)$$

For an order category $i \in \mathcal{I}_{\tilde{w}} \cup \mathcal{M}_{\tilde{w}}$

$$J^* + h(\tilde{w}+1, i) = \bar{h}(\tilde{w})$$

Hence,

$$\begin{aligned} \sum_{j \in \mathcal{J}_{\tilde{w}} \cup \mathcal{K}_{\tilde{w}}} \lambda_j \left(J^* + h(\tilde{w}+1, j)\right) + \sum_{i \in \mathcal{I}_{\tilde{w}} \cup \mathcal{M}_{\tilde{w}}} \lambda_i \left(J^* + h(\tilde{w}+1, i)\right) &= \sum_{j \in \mathcal{J}_{\tilde{w}} \cup \mathcal{K}_{\tilde{w}}} \lambda_j \left(r_j + \bar{h}(\tilde{w} + p_j)\right) + \\ &\quad \sum_{i \in \mathcal{I}_{\tilde{w}} \cup \mathcal{M}_{\tilde{w}}} \lambda_i \bar{h}(\tilde{w}) \end{aligned} \tag{5.19}$$

From their definitions, the intersection of any pair of the sets $\mathcal{I}_{\tilde{w}}$, $\mathcal{J}_{\tilde{w}}$, $\mathcal{K}_{\tilde{w}}$ and $\mathcal{M}_{\tilde{w}}$ is empty. Further every order category belongs to one of the sets $\mathcal{I}_{\tilde{w}}$, $\mathcal{J}_{\tilde{w}}$, $\mathcal{K}_{\tilde{w}}$, $\mathcal{M}_{\tilde{w}}$. Hence $\lambda_{\mathcal{I}_{\tilde{w}}} + \lambda_{\mathcal{J}_{\tilde{w}}} + \lambda_{\mathcal{K}_{\tilde{w}}} + \lambda_{\mathcal{M}_{\tilde{w}}} = 1$. This, together with equation (5.19) yields the following relation.

$$J^* + \bar{h}(\tilde{w}+1) = \lambda_{\mathcal{I}_{\tilde{w}}} \bar{h}(\tilde{w}) + \lambda_{\mathcal{M}_{\tilde{w}}} \bar{h}(\tilde{w}) + \sum_{j \in \mathcal{J}_{\tilde{w}}} \lambda_j \left(r_j + \bar{h}(\tilde{w} + p_j)\right) + \sum_{k \in \mathcal{K}_{\tilde{w}}} \lambda_k \left(r_k + \bar{h}(\tilde{w} + p_k)\right) \tag{5.20}$$

Using similar arguments,

$$J^* + \bar{h}(\tilde{w}) = \lambda_{\mathcal{I}_{\tilde{w}}} \bar{h}(\tilde{w}-1) + \lambda_{\mathcal{K}_{\tilde{w}}} \bar{h}(\tilde{w}-1) + \sum_{j \in \mathcal{J}_{\tilde{w}}} \lambda_j \left(r_j + \bar{h}(\tilde{w} + p_j - 1)\right) + \sum_{m \in \mathcal{M}_{\tilde{w}}} \lambda_m \left(r_m + \bar{h}(\tilde{w} + p_m - 1)\right) \tag{5.21}$$

Subtracting (5.20) from (5.21),

$$
\begin{aligned}
\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1) &= \lambda_{\mathcal{I}_{\tilde{w}}} \left(\bar{h}(\tilde{w}-1) - \bar{h}(\tilde{w})\right) + \sum_{m \in \mathcal{M}_{\tilde{w}}} \lambda_m \left(r_m + \bar{h}(\tilde{w}+p_m-1) - \bar{h}(\tilde{w})\right) \\
&\quad + \sum_{k \in \mathcal{K}_{\tilde{w}}} \lambda_k \left(\bar{h}(\tilde{w}-1) - (r_k + h(\tilde{w}+p_k))\right) + \sum_{j \in \mathcal{J}_{\tilde{w}}} \lambda_j \left(\bar{h}(\tilde{w}+p_j-1) - \bar{h}(\tilde{w}+p_j)\right) \\
&\geq (\lambda_{\mathcal{I}_{\tilde{w}}} + \lambda_{\mathcal{M}_{\tilde{w}}}) \left(\bar{h}(\tilde{w}-1) - \bar{h}(\tilde{w})\right) + \lambda_{\mathcal{J}_{\tilde{w}}} \left(\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1)\right) \\
&\quad + \sum_{k \in \mathcal{K}_{\tilde{w}}} \lambda_k \left(\bar{h}(\tilde{w}-1) - (r_k + \bar{h}(\tilde{w}+p_k-1)) + \bar{h}(\tilde{w}+p_k-1) - \bar{h}(\tilde{w}+p_k)\right) \\
&\geq (\lambda_{\mathcal{I}_{\tilde{w}}} + \lambda_{\mathcal{M}_{\tilde{w}}}) \left(\bar{h}(\tilde{w}-1) - \bar{h}(\tilde{w})\right) + (\lambda_{\mathcal{K}_{\tilde{w}}} + \lambda_{\mathcal{J}_{\tilde{w}}}) \left(\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1)\right) \qquad (5.22)
\end{aligned}
$$

The first inequality comes from the fact that by definition, for $m \in \mathcal{M}_{\tilde{w}}$, $r_m + h(\tilde{w}+p_m-1) \geq \bar{h}(\tilde{w}-1)$ and the assumption that $\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1) \leq \bar{h}(\tilde{w}+1) - \bar{h}(\tilde{w}+2) \leq \cdots \leq \bar{h}(l-2) - \bar{h}(l-1)$. The second inequality comes from the fact that for $k \in \mathcal{K}_{\tilde{w}}$, $\bar{h}(\tilde{w}-1) \geq r_k + \bar{h}(\tilde{w}+p_k-1)$ and also from the assumption that $\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1) \leq \bar{h}(\tilde{w}+1) - \bar{h}(\tilde{w}+2) \leq \cdots \leq \bar{h}(l-2) - \bar{h}(l-1)$. From (5.22),

$$
(1 - \lambda_{\mathcal{J}_{\tilde{w}}} - \lambda_{\mathcal{K}_{\tilde{w}}}) \left(\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1)\right) \geq (\lambda_{\mathcal{I}_{\tilde{w}}} + \lambda_{\mathcal{M}_{\tilde{w}}}) \left(\bar{h}(\tilde{w}-1) - \bar{h}(\tilde{w})\right)
$$

Hence $\bar{h}(\tilde{w}) - \bar{h}(\tilde{w}+1) \geq \bar{h}(\tilde{w}-1) - \bar{h}(\tilde{w})$ since $\lambda_{\mathcal{I}_{\tilde{w}}} + \lambda_{\mathcal{M}_{\tilde{w}}} + \lambda_{\mathcal{J}_{\tilde{w}}} + \lambda_{\mathcal{K}_{\tilde{w}}} = 1$ and $\lambda_{\mathcal{I}_{\tilde{w}}} > 0$. This proves the first part of the lemma.

Arguments similar to those used for establishing (5.20) and (5.21) yield:

$$
\begin{aligned}
J^* + \bar{h}(1) &= \lambda_{\mathcal{I}_0} \bar{h}(0) + \lambda_{\mathcal{M}_0} \bar{h}(0) + \sum_{j \in \mathcal{J}_0} \lambda_j \left(r_j + \bar{h}(p_j)\right) + \sum_{k \in \mathcal{K}_0} \lambda_k \left(r_k + \bar{h}(p_k)\right) \qquad (5.23) \\
J^* + \bar{h}(0) &= \lambda_{\mathcal{I}_0} \bar{h}(0) + \lambda_{\mathcal{K}_0} \bar{h}(0) + \sum_{j \in \mathcal{J}_0} \lambda_j \left(r_j + \bar{h}(p_j-1)\right) \\
&\quad + \sum_{m \in \mathcal{M}_0} \lambda_m \left(r_m + \bar{h}(p_m-1)\right) \qquad (5.24)
\end{aligned}
$$

Subtracting (5.23) from (5.24) and repeating the arguments used in the proof of the first part of the lemma,

$$(1 - \lambda_{\mathcal{J}_0} - \lambda_{\mathcal{K}_0}) \left(\bar{h}(0) - \bar{h}(1)\right) \geq (\lambda_{\mathcal{I}_0} + \lambda_{\mathcal{M}_0}) \left(\bar{h}(0) - \bar{h}(0)\right) = 0$$

The above inequality together with the facts that $\lambda_{\mathcal{I}_0} + \lambda_{\mathcal{M}_0} + \lambda_{\mathcal{J}_0} + \lambda_{\mathcal{K}_0} = 1$ and $\lambda_{\mathcal{I}_0} > 0$, imply $\bar{h}(0) - \bar{h}(1) \geq 0$. From the first part of the lemma, for any $1 < w \leq l - 1$, we have, $\bar{h}(0) - \bar{h}(1) \leq \bar{h}(1) - \bar{h}(2) \leq \cdots \leq \bar{h}(w-1) - \bar{h}(w)$. This, along with the fact that $\bar{h}(0) - \bar{h}(1) \geq 0$, proves the second part of the lemma. □

Consider a state $(\tilde{w}, \bar{i})$ with $\tilde{w} \geq 1$ such that it is optimal to accept an order of category $\bar{i}$. From the Bellman equation for the state $(\tilde{w}, \bar{i})$,

$$J^* + h(\tilde{w}, \bar{i}) = r_{\bar{i}} + \bar{h}(\tilde{w} + p_{\bar{i}} - 1) \geq \bar{h}(\tilde{w} - 1) \tag{5.25}$$

Suppose $\tilde{w} > 1$. Then,

$$\begin{aligned} r_{\bar{i}} + \bar{h}(\tilde{w} + p_{\bar{i}} - 2) &= r_{\bar{i}} + \bar{h}(\tilde{w} + p_{\bar{i}} - 1) + \left(\bar{h}(\tilde{w} + p_{\bar{i}} - 2) - \bar{h}(\tilde{w} + p_{\bar{i}} - 1)\right) \\ &\geq \bar{h}(\tilde{w} - 1) + \left(\bar{h}(\tilde{w} - 2) - \bar{h}(\tilde{w} - 1)\right) \\ &= \bar{h}(\tilde{w} - 2) \end{aligned}$$

The inequality is due to Lemma (5.2.2) and equation (5.25). Accepting an order of category $\bar{i}$ at state $(\tilde{w} - 1, \bar{i})$ is optimal if $r_{\bar{i}} + \bar{h}(\tilde{w} + p_{\bar{i}} - 2) \geq \bar{h}(\tilde{w} - 2)$. Therefore it is optimal to accept an order of category $\bar{i}$ at state $(\tilde{w} - 1, \bar{i})$ if it is optimal to accept an order at state $(\tilde{w}, \bar{i})$ for $\tilde{w} > 1$. Thus if it is optimal to accept an order of category $\bar{i}$ at state $(\tilde{w}, \bar{i})$, then it is also optimal to accept an order from category $\bar{i}$ at states $(w, \bar{i})$ where $1 \leq w \leq \tilde{w}$.

To complete the proof of the theorem we only have to show that it is optimal to accept an order of category $\bar{i}$ at state $(0, \bar{i})$ if it is optimal to accept an order of category $\bar{i}$ at some state $(\tilde{w}, \bar{i})$ where $\tilde{w} \geq 1$. Since it is optimal to accept an order of category $\bar{i}$ at state $(\tilde{w}, \bar{i})$, it is feasible to accept an order of category $\bar{i}$ at state $(1, \bar{i})$.

$$\begin{aligned} r_{\bar{i}} + \bar{h}(p_{\bar{i}} - 1) &= r_{\bar{i}} + \bar{h}(p_{\bar{i}}) + \bar{h}(p_{\bar{i}} - 1) - \bar{h}(p_{\bar{i}}) \\ &\geq \bar{h}(0) + \bar{h}(p_{\bar{i}} - 1) - \bar{h}(p_{\bar{i}}) \\ &\geq \bar{h}(0) \end{aligned}$$

The inequality follows from $r_{\bar{i}} + \bar{h}(p_{\bar{i}}) \geq \bar{h}(0)$ since it is optimal to accept an order of category $\bar{i}$ at state $(1, \bar{i})$. The second inequality comes from the second part of Lemma (5.2.2). Thus if it is optimal to accept an order of category $\bar{i}$ at state $(\tilde{w}, \bar{i})$, it is optimal to accept orders from category $\bar{i}$ for all states $(w, \bar{i})$, with $0 \leq w \leq \tilde{w}$. This completes the proof of the theorem □

Theorem (5.2.1) establishes the existence of thresholds $w_1, w_2, \ldots, w_n$ such that it is optimal to reject an order of category i at states (w, i) where $w_i \leq w \leq l - 1$ for $i \in \mathcal{C}$. It is reasonable to expect that the reward for an order increases with the processing time needed for it. However the growth in the reward may be less than linear with respect to the processing time needed. In such cases, the following proposition provides further insight into the structure of the optimal policy.

Proposition 5.2.2. *Suppose $\frac{r_1}{p_1} \geq \frac{r_2}{p_2} \geq \cdots \geq \frac{r_n}{p_n}$ where $p_1 \leq p_2 \leq \cdots \leq p_n$. If it is optimal to accept an order of category $\bar{k}$ at state $(\tilde{w}, \bar{k})$, it is also optimal to accept all feasible orders at state $(\tilde{w}, k)$ where $1 \leq k \leq \bar{k}$.*

Proof. Suppose $\tilde{w} > 0$. Consider some k such that $1 \leq k < \bar{k}$ such that it is feasible to accept an order of category k. We have,

$$\begin{aligned}
\bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w}+p_{\bar{k}}-1) &= \bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w}+p_k-1)+\bar{h}(\tilde{w}+p_k-1)-\bar{h}(\tilde{w}+p_{\bar{k}}-1)\\
&\geq h(\tilde{w}-1)-\bar{h}(\tilde{w}+p_k-1)+(p_{\bar{k}}-p_k)\left(\bar{h}(\tilde{w}+p_k-1)-\bar{h}(\tilde{w}+p_k)\right)\\
&\geq h(\tilde{w}-1)-\bar{h}(\tilde{w}+p_k-1)+(p_{\bar{k}}-p_k)\left(\frac{h(\tilde{w}-1)-\bar{h}(\tilde{w}+p_k-1)}{p_k}\right)\\
&= \frac{p_{\bar{k}}}{p_k}\left((h(\tilde{w}-1)-\bar{h}(\tilde{w}+p_k-1)\right) \qquad (5.26)
\end{aligned}$$

The first and the second inequalities come from Lemma (5.2.2). Since it is optimal to accept an order of category $\bar{k}$ at state $(\tilde{w},\bar{k})$, from the optimality condition for the state $(\tilde{w},\bar{k})$, we have, $r_{\bar{k}}+\bar{h}(\tilde{w}+p_{\bar{k}}-1)\geq\bar{h}(\tilde{w}-1)$ which can be restated as, $\frac{r_{\bar{k}}}{p_{\bar{k}}}\geq\frac{\bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w}+p_{\bar{k}}-1)}{p_{\bar{k}}}$. Thus,

$$\frac{r_k}{p_k}\geq\frac{r_{\bar{k}}}{p_{\bar{k}}}\geq\frac{\bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w}+p_{\bar{k}}-1)}{p_{\bar{k}}}\geq\frac{\bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w}+p_k-1)}{p_k}$$

The last inequality comes from (5.26). Hence $r_k+\bar{h}(\tilde{w}+p_k-1)\geq\bar{h}(\tilde{w}-1)$ establishing the optimality of accepting an order of category k at state $(\tilde{w},k)$.

Suppose $\tilde{w}=0$. Since $p_k\leq p_{\bar{k}}$, $\frac{p_k}{p_k-1}\geq\frac{p_{\bar{k}}}{p_{\bar{k}}-1}$ and hence

$$\frac{r_k}{p_k-1}=\frac{r_k}{p_k}\frac{p_k}{p_k-1}\geq\frac{r_{\bar{k}}}{p_{\bar{k}}}\frac{p_{\bar{k}}}{p_{\bar{k}}-1}\geq\frac{r_{\bar{k}}}{p_{\bar{k}}-1}\geq\frac{\bar{h}(0)-\bar{h}(p_{\bar{k}}-1)}{p_{\bar{k}}-1}\geq\frac{\bar{h}(0)-\bar{h}(p_k-1)}{p_k-1}$$

Thus $\frac{r_k}{p_k-1}\geq\frac{\bar{h}(0)-\bar{h}(p_k-1)}{p_k-1}$ establishing the optimality of accepting an order of category k at $(0,k)$

5.2.4 Extensions

We established the structure of the optimal policy for problem (5.13). We now show that a similar structure exists for the optimal policy for two other closely related problems.

5.2.5 Order rejection penalty

We now consider an order acceptance problem with Bernoulli arrival process where there is a category dependent penalty for rejecting an arriving order. We refer to this problem as (OAP-B-RP). In this case for every order category $i \in C$, there is an additional parameter, the rejection penalty denoted by c_i. This quantifies the loss to the firm due to the rejection of an order from that category. We assume that the FCFS based scheduling policy is used for accepted orders. Since this additional parameter affects only the reward structure for the problem, the state space and the transition structure remain the same as in Section 5.2. While Lemma 5.2.1 need not hold for this problem, Theorem 5.2.1 holds for this problem. As before, we let the order category 0 denote the situation when no order has arrived during a time period.

Theorem 5.2.2.

Let c_i be the loss due to the rejection of an order of category i for $i \in C$ and let $c_0 = 0$. Suppose it is optimal to accept an order of category $\bar{i}$ at state $(\bar{w}, \bar{i})$. Then it is also optimal to accept an order of category $\bar{i}$ at all states $(w, \bar{i})$ with $w < \bar{w}$.

Proof. The proof is similar to the proof of Theorem 5.2.1. We refer to the proof of Theorem 5.2.1 frequently to avoid the repetition of arguments. Lemma 5.2.2 holds for this problem and we state and prove it for completeness. As before, $l = max_{j \in C}\{l_j\}$ and for $w = 0, 1, \ldots, l-1$, $\bar{h}(w) = \sum_{j=0}^{n} \lambda_j h(w, j)$.

Lemma 5.2.3. *Let h be an optimal differential value function for OAP-B-RP with FCFS based scheduling of accepted orders.*

1. *For* $0 < w < l-1$, $\bar{h}(w) - \bar{h}(w+1) \geq \bar{h}(w-1) - \bar{h}(w)$.

2. *For* $0 < w \leq l-1$, $\bar{h}(w-1) - \bar{h}(w) \geq 0$

Proof. The proof is similar to the proof of Lemma 5.2.2. We arrange the order categories so that $p_1 \leq \cdots \leq p_n$. For $l - p_1 + 1 \leq w \leq l - 1$, the Bellman's equation for the state (w, i) is

$$J^* + h(w, i) = \bar{h}(w - 1) - c_i \tag{5.27}$$

Thus, $\bar{h}(w - 1) = \bar{h}(w) + J^* + \sum_{j=0}^{n} \lambda_i c_i$ for $l - p_1 + 1 \leq w \leq l - 1$. Since $p_1 > 2$ by assumption, $\bar{h}(l - 2) - \bar{h}(l - 1) = \bar{h}(l - 3) - \bar{h}(l - 2)$ and thus the lemma is true for $w = l - 2$. We now assume that the lemma is true for $l - 2 \geq w \geq \tilde{w} + 1$ where $\tilde{w} > 0$. That is, $\bar{h}(w) - \bar{h}(w + 1) \geq \bar{h}(w - 1) - \bar{h}(w)$ for $l - 2 \geq w \geq \tilde{w} + 1$. We show that the lemma is true for $\tilde{w}$, that is, $\bar{h}(\tilde{w}) - \bar{h}(\tilde{w} + 1) \geq \bar{h}(\tilde{w} - 1) - \bar{h}(\tilde{w})$. This would complete the proof for the first part of the lemma.

For w, such that $0 \leq w \leq l - 2$, we define the sets $\mathcal{I}_w$, $\mathcal{J}_w$, $\mathcal{K}_w$, $\mathcal{M}_w$ and the quantities, $\lambda_{\mathcal{I}_w}$,$\lambda_{\mathcal{J}_w}$, $\lambda_{\mathcal{K}_w}$, $\lambda_{\mathcal{M}_w}$ as in the proof of Lemma 5.2.2. The optimality equation for state $(\tilde{w} + 1, j)$ with $j \in \mathcal{J}_{\tilde{w}} \cup \mathcal{K}_{\tilde{w}}$

$$J^* + h(\tilde{w} + 1, j) = r_j + \bar{h}(\tilde{w} + p_j)$$

The optimality equation for state $(\tilde{w} + 1, i)$ with $i \in \mathcal{I}_{\tilde{w}} \cup \mathcal{M}_{\tilde{w}}$

$$J^* + h(\tilde{w} + 1, i) = \bar{h}(\tilde{w}) - c_i$$

Using arguments similar to the proof of Lemma 5.2.2, we obtain the following relations

$$\begin{aligned} J^* + \bar{h}(\tilde{w} + 1) &= \lambda_{\mathcal{I}_{\tilde{w}}} \bar{h}(\tilde{w}) + \lambda_{\mathcal{M}_{\tilde{w}}} \bar{h}(\tilde{w}) - \sum_{i \in \mathcal{I}_{\tilde{w}}} \lambda_i c_i - \sum_{m \in \mathcal{M}_{\tilde{w}}} \lambda_m c_m \\ &\quad + \sum_{j \in \mathcal{J}_{\tilde{w}}} \lambda_j \left(r_j + \bar{h}(\tilde{w} + p_j)\right) + \sum_{k \in \mathcal{K}_{\tilde{w}}} \lambda_k \left(r_k + \bar{h}(\tilde{w} + p_k)\right) \qquad (5.28) \\ J^* + \bar{h}(\tilde{w}) &= \lambda_{\mathcal{I}_{\tilde{w}}} \bar{h}(\tilde{w} - 1) + \lambda_{\mathcal{K}_{\tilde{w}}} \bar{h}(\tilde{w} - 1) - \sum_{i \in \mathcal{I}_{\tilde{w}}} \lambda_i c_i - \sum_{k \in \mathcal{K}_{\tilde{w}}} \lambda_k c_k + \\ &\quad \sum_{j \in \mathcal{J}_{\tilde{w}}} \lambda_j \left(r_j + \bar{h}(\tilde{w} + p_j - 1)\right) + \sum_{m \in \mathcal{M}_{\tilde{w}}} \lambda_m \left(r_m + \bar{h}(\tilde{w} + p_m - 1)\right) \qquad (5.29) \end{aligned}$$

Subtracting (5.28) from (5.29) and using arguments similar to those used in the derivation of equation (5.22) in the proof of Lemma 5.2.2 establishes the following relation

$$\bar{h}(\tilde{w})-\bar{h}(\tilde{w}+1) \geq (\lambda_{\mathcal{I}_{\tilde{w}}}+\lambda_{\mathcal{M}_{\tilde{w}}})\left(\bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w})\right)+(\lambda_{\mathcal{K}_{\tilde{w}}}+\lambda_{\mathcal{J}_{\tilde{w}}})\left(\bar{h}(\tilde{w})-\bar{h}(\tilde{w}+1)\right) \quad (5.30)$$

Rearrangement of (5.30) together with the fact that $\lambda_{\mathcal{I}_{\tilde{w}}} > 0$ proves the first part of the lemma.

Let $\mathcal{I}_0$, $\mathcal{J}_0$, $\mathcal{K}_0$, $\mathcal{M}_0$ be defined as in the proof of Lemma 5.2.2. Using arguments similar to those used in the proof of the first part of the Lemma, it can be established that

$$\begin{aligned} J^* + \bar{h}(1) &= \lambda_{\mathcal{I}_0}\bar{h}(0) + \lambda_{\mathcal{M}_0}\bar{h}(0) - \sum_{i\in\mathcal{I}_0}\lambda_i c_i - \sum_{m\in\mathcal{M}_0}\lambda_m c_m \\ &\quad + \sum_{j\in\mathcal{J}_0}\lambda_j\left(r_j+\bar{h}(p_j)\right) + \sum_{k\in\mathcal{K}_0}\lambda_k\left(r_k+\bar{h}(p_k)\right) \end{aligned} \quad (5.31)$$

$$\begin{aligned} J^* + \bar{h}(0) &= \lambda_{\mathcal{I}_0}\bar{h}(0) + \lambda_{\mathcal{K}_0}\bar{h}(0) - \sum_{i\in\mathcal{I}_0}\lambda_i c_i - \sum_{k\in\mathcal{K}_0}\lambda_k c_k \\ &\quad + \sum_{j\in\mathcal{J}_0}\lambda_j\left(r_j+\bar{h}(p_j-1)\right) + \sum_{m\in\mathcal{M}_0}\lambda_m\left(r_m+\bar{h}(p_m-1)\right) \end{aligned} \quad (5.32)$$

Subtracting (5.31) from (5.32) and repeating the arguments used in the proof of the first part of the lemma,

$$(1-\lambda_{\mathcal{J}_0}-\lambda_{\mathcal{K}_0})\left(\bar{h}(0)-\bar{h}(1)\right) \geq (\lambda_{\mathcal{I}_0}+\lambda_{\mathcal{M}_0})\left(\bar{h}(0)-\bar{h}(0)\right)$$

Using the above equation and the fact that $\lambda_{\mathcal{I}_0}+\lambda_{\mathcal{M}_0}+\lambda_{\mathcal{J}_0}+\lambda_{\mathcal{K}_0}=1$ and $\lambda_{\mathcal{I}_0}>0$, we conclude that $\bar{h}(0)-\bar{h}(1)\geq 0$. From the first part of the lemma for $1 < w \leq l-1$, we have, $\bar{h}(0)-\bar{h}(1) \leq \bar{h}(1)-\bar{h}(2) \leq \cdots \leq \bar{h}(w-1)-\bar{h}(w)$. Together with the fact that $\bar{h}(0)-\bar{h}(1)\geq 0$, this proves the second part of the lemma. □

Consider some state $(\tilde{w}, \bar{i})$, such that it is optimal to accept an order of category $\bar{i}$ at the state. If $\tilde{w} > 0$, from the optimality equation for this state, we have

$$r_{\bar{i}} + \bar{h}(\tilde{w} + p_{\bar{i}} - 1) \geq \bar{h}(\tilde{w} - 1) - c_{\bar{i}} \tag{5.33}$$

If $\tilde{w} = 0$, the optimality equation for this state yields the relation,

$$r_{\bar{i}} + \bar{h}(p_{\bar{i}} - 1) \geq \bar{h}(0) - c_{\bar{i}} \tag{5.34}$$

Using arguments similar to the proof of Theorem 5.2.1 following the proof of Lemma (5.2.2) together with equations (5.33) and (5.34) proves the theorem.

5.2.6 Reward and lead time quotation

In this section, we consider a problem where the decision to be made each time is to reliably quote a reward, lead time pair for arriving orders. We assume that the orders arrive via a Bernoulli process similar to problem OAP-B and we also assume that each arriving order can be belong to one of n categories. As before we let $\mathcal{C}$ denote the set $\{1, \ldots, n\}$ and we let $\bar{\mathcal{C}}$ denote the set $\{0, 1, \ldots, n\}$. Associated with order category i, $i \in \mathcal{C}$ is a deterministic processing time p_i and a set of ordered pairs, $\mathcal{A}_i = \{(r_i^1, l_i^1), \ldots, (r_i^{k_i}, l_i^{k_i})\}$. Every element of the set $\mathcal{A}_i$ consists of a reward , lead time pair which when quoted would be accepted by the customer. For an order category i, $i \in \mathcal{C}$, there are k_i such acceptable reward, lead time pairs. We assume that the firm can effectively reject an arriving order of category i, $i \in \mathcal{C}$ by quoting an appropriate reward, lead time pair that does not belong to $\mathcal{A}_i$. We assume without loss of generality that $l_i^1 < l_i^2 < \cdots < l_i^{k_i}$ for $i \in \mathcal{C}$. We also assume that $r_i^1 > r_i^2 > \cdots > r_i^{k_i}$. In words, this assumption means that for a particular order category the reward to the firm from an order decreases with increasing lead time. Once an order arrives, the firm has to decide whether to accept the order and if so the reward, lead time pair to be quoted. We make the simplification that accepted orders are scheduled on a FCFS basis and we establish a structural result on the optimal policy reward, lead time quotation policy under this assumption.

The state of the system at time t, under FCFS scheduling of accepted orders can be represented by $x_t = (w^t, j^t)$ where w^t is the remaining work at time t and j^t is the category of the order arriving at time t. Consider some state $x = (w, i)$. We observe that the set of feasible reward, lead time pairs at state x is $\{(r_i^\tau, l_i^\tau); w + p_i \leq l_i^\tau; 1 \leq \tau \leq k_i\}$. We represent the set of feasible reward, lead time pairs at state $x = (w, i)$ by $\mathcal{A}_x = \{\tau; w + p_i \leq l_i^\tau\}$. Let l denote the highest lead time that can be feasibly quoted for any order category. That is, $l = \max_{i \in C}(l_i^{k_i})$.

We now describe the state transition structure for this problem. Suppose the state of the system is (w, i), $i \in C$ and a reward, lead time pair of $(r_i^m, l_i^m) \in A_i$ is quoted. The system transitions to the state $(w + p_i - 1, j)$ with a probability λ_j for $j \in \bar{C}$. Suppose the state of the system is (w, i) , $i \in \bar{C}$. If the order is rejected then the system transitions to the state $(w - 1, j)$ with a probability λ_j for $j \in \bar{C}$ if $w \neq 0$. If $w = 0$, then the system transitions to the state $(0, j)$ with a probability λ_j for $j \in \bar{C}$.

It can be verified that the state $(0, 0)$ satisfies the assumption in the proposition 5.1.1 and so there is a single recurrent class for the Markov chain induced by any stationary policy. Thus condition (1) of proposition 4.2.6 of [8] is satisfied and there exists a solution to the Bellman equation for this problem. We now state the version of theorem 5.2.1 for this problem.

Theorem 5.2.3. 1. *Suppose it is optimal to accept an order of category $\bar{i}$ at state $(\bar{w}, \bar{i})$. Then it is optimal to quote the reward, lead time pair with the smallest feasible lead time.*

2. *Suppose it is optimal to accept an order of category $\bar{i}$ at state $(\bar{w}, \bar{i})$. Then it is also optimal to accept an order of category $\bar{i}$ at all states $(w, \bar{i})$ with $w < \bar{w}$.*

proof of 1

Suppose $\bar{w} > 0$. The optimality equation at state $\bar{x} = (\bar{w}, \bar{i})$ is,

$$J^* + h(\bar{w}, \bar{i}) = max\{max_{j \in \mathcal{A}_{\bar{x}}}\{r_{\bar{i}}^j + \bar{h}(\bar{w} + p_{\bar{i}} - 1)\}, \bar{h}(\bar{w} - 1)\} \tag{5.35}$$

where h is an optimal differential value function for this problem and $\bar{h}(w) = \sum_{j=1}^n \lambda_j h(w, j)$ and J^* is the optimal expected average reward per time as usual. In the following, we define i_w as the index of the reward, lead time pair for order category i that has the smallest feasible lead time when the remaining work is w. Hence $w + p_i \leq l_i^\tau$ if and only if $k_i \geq \tau \geq i_w$. Suppose it is optimal to accept an order of category $\bar{i}$ at state $(\bar{w}, \bar{i})$, then $max_{\tau \in \mathcal{A}_{\bar{x}}}\{r_{\bar{i}}^\tau + \bar{h}(\bar{w} + p_{\bar{i}} - 1)\} \geq \bar{h}(\bar{w} - 1)$. The reward, lead time pair that attains the maximum in $max_{\tau \in \mathcal{A}_{\bar{x}}}\{r_{\bar{i}}^\tau + \bar{h}(\bar{w} + p_i - 1)\}$ is $\bar{i}_w$ since $r_{\bar{i}}^\tau \leq r_{\bar{i}}^{\bar{i}_w}$ for all $\tau \in \mathcal{A}_{\bar{x}}$ due to the assumption that the reward decreases with increasing lead time. The proof is similar when $\bar{w} = 0$.

proof of 2

The proof of the second part of the theorem is very similar to the proof of theorem 5.2.1. We refer to the proof of theorem 5.2.1 frequently to avoid the repetition of arguments. Lemma 5.2.2 holds for this problem as well and we state and prove it for completeness.

Lemma 5.2.4. *1. For* $0 < w < l - 1$, $\bar{h}(w) - \bar{h}(w+1) \geq \bar{h}(w-1) - \bar{h}(w)$.

2. For $0 < w \leq l - 1$, $\bar{h}(w-1) - \bar{h}(w) \geq 0$

Proof. As in the proof of Lemma 5.2.2 we arrange the order categories so that $p_1 \leq \cdots \leq p_n$. For $l - p_1 + 1 \leq w \leq l - 1$, the Bellman's equation for the state (w, i) is

$$J^* + h(w, i) = \bar{h}(w - 1) \tag{5.36}$$

Thus, $\bar{h}(w-1) = \bar{h}(w) + J^*$ for $l - 1 \geq w \geq l - p_1 + 1$. Since $p_1 > 2$ by assumption, $\bar{h}(l-2) - \bar{h}(l-1) = \bar{h}(l-3) - \bar{h}(l-2)$ and thus the lemma is true for $w = l - 2$. We now assume that the lemma is true for $l-2 \geq w \geq \tilde{w}+1$ where $\tilde{w} > 0$. Under this assumption, we show that the lemma is true for $\tilde{w}$ which would complete the proof of the first part of the lemma.

For a given w such that $0 \leq w \leq l-2$, we define the sets $\mathcal{I}_w$, $\mathcal{J}_w$, $\mathcal{K}_w$, $\mathcal{M}_w$ and $\lambda_{\mathcal{I}_w}$,$\lambda_{\mathcal{J}_w}$, $\lambda_{\mathcal{K}_w}$, $\lambda_{\mathcal{M}_w}$ as in the proof of Lemma 5.2.2. We have already shown in the first part of the theorem that if it is optimal to accept an order of category i at state (w,i), then $(r_i^{i_w}, l_i^{i_w})$ is the quoted reward, lead time pair. Therefore, optimality equation for state $(\tilde{w}+1, j)$ with $j \in \mathcal{J}_{\tilde{w}} \cup \mathcal{K}_{\tilde{w}}$ is

$$J^* + h(\tilde{w}+1, j) = r_j^{j_{\tilde{w}+1}} + \bar{h}(\tilde{w}+p_j)$$

The optimality equation for state $(\tilde{w}+1, i)$ with $i \in \mathcal{I}_{\tilde{w}} \cup \mathcal{M}_{\tilde{w}}$

$$J^* + h(\tilde{w}+1, i) = \bar{h}(\tilde{w})$$

Using arguments similar to the proof of Lemma 5.2.2, we obtain the following relation

$$\begin{aligned}
J^* + \bar{h}(\tilde{w}+1) &= \lambda_{\mathcal{I}_{\tilde{w}}}\bar{h}(\tilde{w}) + \lambda_{\mathcal{M}_{\tilde{w}}}\bar{h}(\tilde{w}) + \sum_{j\in\mathcal{J}_{\tilde{w}}} \lambda_j \left(r_j^{j_{\tilde{w}+1}} + \bar{h}(\tilde{w}+p_j)\right) + \\
&\quad \sum_{k\in\mathcal{K}_{\tilde{w}}} \lambda_k \left(r_k^{k_{\tilde{w}+1}} + \bar{h}(\tilde{w}+p_k)\right) \qquad (5.37)\\
J^* + \bar{h}(\tilde{w}) &= \lambda_{\mathcal{I}_{\tilde{w}}}\bar{h}(\tilde{w}-1) + \lambda_{\mathcal{K}_{\tilde{w}}}\bar{h}(\tilde{w}-1) + \sum_{j\in\mathcal{J}_{\tilde{w}}} \lambda_j \left(r_j^{j_{\tilde{w}}} + \bar{h}(\tilde{w}+p_j-1)\right) + \\
&\quad \sum_{m\in\mathcal{M}_{\tilde{w}}} \lambda_m \left(r_m^{m_{\tilde{w}}} + \bar{h}(\tilde{w}+p_m-1)\right) \qquad (5.38)
\end{aligned}$$

A consequence of the first part of the theorem and the assumption that the quoted reward decreases with increasing lead time is that $r_q^{q_{\tilde{w}+1}} \leq r_q^{q_{\tilde{w}}}$ for $q \in \mathcal{C}$. Subtracting (5.37) from (5.38) and using arguments similar to those used in the derivation of equation (5.22) in the proof of Lemma 5.2.2 and also using the fact $r_j^{j_{\tilde{w}+1}} \leq r_j^{j_{\tilde{w}}}, j \in \mathcal{J}_{\tilde{w}}$ and $r_k^{k_{\tilde{w}+1}} \leq r_k^{k_{\tilde{w}}}, k \in \mathcal{K}_{\tilde{w}}$ establishes the following relation

$$\bar{h}(\tilde{w})-\bar{h}(\tilde{w}+1) \geq \left(\lambda_{\mathcal{I}_{\tilde{w}}}+\lambda_{\mathcal{M}_{\tilde{w}}}\right)\left(\bar{h}(\tilde{w}-1)-\bar{h}(\tilde{w})\right)+\left(\lambda_{\mathcal{K}_{\tilde{w}}}+\lambda_{\mathcal{J}_{\tilde{w}}}\right)\left(\bar{h}(\tilde{w})-\bar{h}(\tilde{w}+1)\right) \quad (5.39)$$

Rearrangement of (5.39)and using the fact that $\lambda_{\mathcal{I}_{\tilde{w}}} > 0$, proves the first part of the lemma.

Using arguments similar to those used in the proof of the first part of the lemma, it can be established that

$$\begin{aligned} J^*+\bar{h}(1) &= \lambda_{\mathcal{I}_0}\bar{h}(0)+\lambda_{\mathcal{M}_0}\bar{h}(0)+\sum_{j\in\mathcal{J}_0}\lambda_j\left(r_j^{j_1}+\bar{h}(p_j)\right)+\sum_{k\in\mathcal{K}_0}\lambda_k\left(r_k^{k_1}+\bar{h}(p_k)\right) \quad (5.40)\\ J^*+\bar{h}(0) &= \lambda_{\mathcal{I}_0}\bar{h}(0)+\lambda_{\mathcal{K}_0}\bar{h}(0)+\sum_{j\in\mathcal{J}_0}\lambda_j\left(r_j^{j_0}+\bar{h}(p_j-1)\right)+\\ &\quad \sum_{m\in\mathcal{M}_0}\lambda_m\left(r_m^{m_0}+\bar{h}(p_m-1)\right) \quad (5.41)\end{aligned}$$

Subtracting (5.40) from (5.41) and repeating the arguments used in the proof of the first part of the lemma,

$$(1-\lambda_{\mathcal{J}_0}-\lambda_{\mathcal{K}_0})\left(\bar{h}(0)-\bar{h}(1)\right) \geq (\lambda_{\mathcal{I}_0}+\lambda_{\mathcal{M}_0})\left(\bar{h}(0)-\bar{h}(0)\right)$$

Using the above equation and the fact that $\lambda_{\mathcal{I}_0}+\lambda_{\mathcal{M}_0}+\lambda_{\mathcal{J}_0}+\lambda_{\mathcal{K}_0} = 1$ and $\lambda_{\mathcal{I}_0} > 0$, $\bar{h}(0)-\bar{h}(1) \geq 0$. From the first part of the lemma for any $1 < w \leq l-1$, we have, $\bar{h}(0)-\bar{h}(1) \leq \bar{h}(1)-\bar{h}(2) \leq \cdots \leq \bar{h}(w-1)-\bar{h}(w)$. Together with the fact that $\bar{h}(0)-\bar{h}(1) \geq 0$, this proves the second part of the lemma. □

It can be seen that the arguments in the proof of theorem 5.2.1 following the proof of Lemma 5.2.2 can be used together with the fact $r_j^{j_{\tilde{w}+1}} \leq r_j^{j_{\tilde{w}}}$ for $0 \leq \tilde{w} \leq l-p_1-1$ and $j \in \mathcal{C}$ to prove the theorem. □

5.3 Expected Average Reward Per Time For An Autonomous Queue With One Order Category

In Section (5.2), we established structural results for OAP-B with FCFS scheduling policy for accepted orders. In this section we consider a MTO manufacturing firm where the orders belong to only one category. In this case every feasible order is accepted and hence the associated queuing system can be considered autonomous with no control. We derive an expression for the expected average reward per time for this system. From this expression we also obtain the steady state probability for the rejection of an order. This probability is referred to as the blocking probability in communications literature. Besides being an interesting queuing system in its own right, the technique we use to derive the expression for the expected average reward per time for this system is also useful in solving a related optimization problem as we will show in the next Section.

Brun *et al.* [10] obtain blocking probabilities for M/D/1/k queues. Gravey *et al.* [26] present closed form expressions for the steady state probabilities for remaining work for a Geo/D/1/k system. Linwong *et al.* [37] present an approximation scheme for computing the buffer size for a given blocking probability for a Geo/D/1/k system under some restrictive assumptions on the arrival rates. The setting we consider is similar to those considered by Gravey *et al.* [26] and Linwong *et al.* [37]. However, we have no restrictions on the arrival rate. Also, since the lead time for the only order category can be an arbitrary integer, the queuing system that we consider is a more general version of the Geo/D/1/k queue. Indeed when the lead time is a integral multiple of the processing time, we have a Geo/D/1/k queue.

We recall that arrival of orders takes place at regularly spaced discrete times and follows a Bernoulli arrival process. We let $j^t = 1$ denote the arrival of an order at time t and we let $j^t = 0$ denote the non-arrival of an order at time t. We let (w^t, j^t) denote the state of the system at time t with w^t being the remaining work. Let l denote the lead time of the only order category and p denote the deterministic processing time for executing the order. Also let r denote the reward obtained from accepting an order. We let λ denote the arrival rate of the orders. Since the autonomous queue is a special case of the more general problem formulated in the previous section, the results from the previous section hold and in particular, the expected average reward per time is the same for all initial states. We denote this common expected average reward per time by J. It is clear that the remaining work w only takes values between 0 and $l-1$ included. For states with remaining work w such that $l-1 \geq w \geq l-p+1$ all orders are rejected since they cannot be delivered by their due date. Hence the Bellman equation for $l-1 \geq w \geq l-p+1$ and for $i = 0, 1$ is,

$$J + h(w, i) = \bar{h}(w-1)$$

where h is a differential value function that satisfies the Bellman equation and for $k = 0, \ldots, l-1$, $\bar{h}(k) = \lambda h(k, 1) + (1-\lambda) h(k, 0)$. For w, where $l-1 \geq w \geq l-p+1$, multiplying the above equation by λ and $1-\lambda$ for $i = 0, 1$ respectively and summing, we get

$$J + \bar{h}(w) = \bar{h}(w-1) \tag{5.42}$$

All orders arriving at states with $0 \leq w \leq l-p$ are accepted since they are feasible. Hence the Bellman equation for $w = l-p$ and $i = 1, 0$ are given by,

$$J + h(l-p, 1) = r + \bar{h}(l-1)$$

and

$$J + h(l-p, 0) = \bar{h}(l-p-1)$$

respectively.

Multiplying the above equations by λ and $(1-\lambda)$ respectively and summing up, we get

$$J + \bar{h}(l-p) = (1-\lambda)\bar{h}(l-p-1) + \lambda(r + \bar{h}(l-1))$$

Since $J = (1-\lambda)J + \lambda J$ and $\bar{h}(l-p) = (1-\lambda)\bar{h}(l-p) + \lambda\bar{h}(l-p)$, we can rewrite the above equation as follows,

$$\begin{aligned}
(1-\lambda)(\bar{h}(l-p-1) - \bar{h}(l-p)) &= (1-\lambda)J + \lambda(J + \bar{h}(l-p)) - \lambda(r + \bar{h}(l-1)) \\
&= (1-\lambda)J - \lambda(r + \bar{h}(l-1) - \bar{h}(l-p) - J) \\
&= (1-\lambda)J - \lambda(r - pJ)
\end{aligned}$$

The last equality follows from (5.42). Hence,

$$\bar{h}(l-p-1) - \bar{h}(l-p) = J - \frac{\lambda(r-pJ)}{1-\lambda} \tag{5.43}$$

Starting similarly from the Bellman equation and rearranging we have for all $1 \leq w \leq l-p$,

$$\bar{h}(w-1) - \bar{h}(w) = J - \frac{\lambda(r + \bar{h}(w+p-1) - \bar{h}(w) - J)}{1-\lambda} \tag{5.44}$$

Define a function f such that $\bar{h}(w-1) - \bar{h}(w) = J - f(l-w)$ for $w = 1, \ldots, l-1$. Equations (5.42) and (5.43) together imply that $f(k) = 0$ for $k = 1, \ldots, p-1$ and $f(p) = \frac{\lambda(r-pJ)}{1-\lambda}$. Substituting the definition for f in (5.44), we have for $1 \leq w \leq l-p$,

$$\begin{aligned}
f(l-w) &= \frac{\lambda(r - J + \bar{h}(w+p-1) - \bar{h}(w))}{1-\lambda} \\
&= \frac{\lambda\left(r - J + \sum_{m=1}^{p-1}(h(w+m) - h(w+m-1)\right)}{1-\lambda} \\
&= \frac{\lambda\left(r - pJ + \sum_{m=1}^{p-1} f(l-w-m)\right)}{1-\lambda} \\
&= \frac{\lambda(r-pJ)}{1-\lambda} + \frac{\lambda\left(\sum_{m=1}^{p-1} f(l-w-m)\right)}{1-\lambda}
\end{aligned} \tag{5.45}$$

For $1 \leq w \leq l-p-1$, the above recursive equation can be simplified as follows,

$$\begin{aligned}
f(l-w) &= \frac{\lambda(r-pJ)}{1-\lambda} + \frac{\lambda\left(f(l-w-p) - f(l-w-p) + \sum_{m=1}^{p-1} f(l-w-m)\right)}{1-\lambda} \\
&= \frac{\lambda(r-pJ)}{1-\lambda} + \frac{\lambda(\sum_{m=1}^{p-1} f(l-w-1-m))}{1-\lambda} + \frac{\lambda(f(l-w-1) - f(l-w-p))}{1-\lambda} \\
&= f(l-w-1) + \frac{\lambda(f(l-w-1) - f(l-w-p))}{1-\lambda} \\
&= \left(1 + \frac{\lambda}{1-\lambda}\right) f(l-w-1) - \frac{\lambda}{1-\lambda} f(l-w-p)
\end{aligned}$$

The third equation comes from (5.45). Rewriting the last equation above using a different index, we have, for $k = p+1, \ldots, l-1$

$$f(k) = \left(1 + \frac{\lambda}{1-\lambda}\right) f(k-1) - \frac{\lambda}{1-\lambda} f(k-p) \tag{5.46}$$

We now derive an expression for $f(k)$ in terms of another function g that depends only on λ, k and p. For convenience let $\lambda_s = \frac{\lambda}{1-\lambda}$.

Proposition 5.3.1. *For $k = 1, 2, \ldots, l-1$, $f(k) = g(k)\lambda_s(r-pJ)$, where $g(k) = 0$ for $k = 1, \ldots, p-1$ and $g(p) = 1$ and $g(k) = (1+\lambda_s)g(k-1) - \lambda_s g(k-p)$ for all $k > p$.*

Proof. The proof of the proposition is by induction on k. From the definition of g and f and (5.43), the validity of the proposition is clear for $k = 1, \ldots, p$.

$$\begin{aligned}
f(p+1) &= (1+\lambda_s)f(p) - \lambda_s f(1) \\
&= (1+\lambda_s)\lambda_s(r-pJ) \\
&= g(p+1)\lambda_s(r-pJ)
\end{aligned}$$

where the first equality follows from equation (5.46). The last equality follows from the fact that by definition, $g(p+1) = (1+\lambda_s)g(p) - \lambda_s g(1) = (1+\lambda_s)$. This proves the proposition for

$k = p + 1$. Suppose the proposition is true for $1, \ldots, \bar{k} - 1$ where $\bar{k} - 1 > p$

$$\begin{aligned}
f(\bar{k}) &= (1+\lambda_s)f(\bar{k}-1) - \lambda_s f(\bar{k}-p) \\
&= (1+\lambda_s)g(\bar{k}-1)\lambda_s(r-pJ) - \lambda_s g(\bar{k}-p)\lambda_s(r-pJ) \\
&= \left((1+\lambda_s)g(\bar{k}-1) - \lambda_s g(\bar{k}-p)\right)\lambda_s(r-pJ) \\
&= g(\bar{k})\lambda_s(r-pJ)
\end{aligned}$$

The first equation is just (5.46) expressed using λ_s. The second equality comes from assumption that the proposition is valid for $\bar{k}-1$ and $\bar{k}-p$ and the last equality comes from the definition of $g(k)$ for $k > p$. □

The following proposition gives a closed form expression for the function g.

Proposition 5.3.2. *Let* $g(k) = 0$ *for* $k = 1, \ldots, p-1$, $g(p) = 1$ *and for* $k > p$, $g(k) = (1 + \lambda_s)g(k-1) - \lambda_s g(k-p)$. *Then* $g(k)$ *is given by the following expression*

$$g(k) = \sum_{j=1}^{\lfloor \frac{k}{p} \rfloor} (-1)^{j-1} \binom{k - jp + j - 1}{j-1} \lambda_s^{j-1} (1+\lambda_s)^{k-jp}$$

Proof. The proof of the proposition is by induction on k. From the definition of g we see that

$$\begin{aligned}
g(p+1) &= (1+\lambda_s)g(p) - \lambda_s g(1) \\
&= (1+\lambda_s)
\end{aligned}$$

The last equality comes from the fact that $g(p) = 1$ and $g(1) = 0$ by definition. This is the same as the expression for $g(p+1)$ in the proposition and thus the proposition is true for $k = p+1$.

Suppose that the proposition is valid for $k = 1, \ldots, v$ where $v > p + 1$. Then by definition,

$$
\begin{aligned}
g(v+1) &= (1+\lambda_s)g(v) - \lambda_s g(v+1-p) \\
&= (1+\lambda_s)\sum_{j=1}^{\lfloor \frac{v}{p} \rfloor}(-1)^{j-1}\binom{v-jp+j-1}{j-1}\lambda_s^{j-1}(1+\lambda_s)^{v-jp} \\
&\quad -\lambda_s \sum_{j=1}^{\lfloor \frac{v+1-p}{p} \rfloor}(-1)^{j-1}\binom{v+1-p-jp+j-1}{j-1}\lambda_s^{j-1}(1+\lambda_s)^{v+1-p-jp} \\
&= \sum_{j=1}^{\lfloor \frac{v}{p} \rfloor}(-1)^{j-1}\binom{v-jp+j-1}{j-1}\lambda_s^{j-1}(1+\lambda_s)^{v+1-jp} \\
&\quad - \sum_{j=1}^{\lfloor \frac{v+1-p}{p} \rfloor}(-1)^{j-1}\binom{v+1-p-jp+j-1}{j-1}\lambda_s^{j}(1+\lambda_s)^{v+1-p-jp} \\
&= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v}{p} \rfloor}(-1)^{j-1}\binom{v-jp+j-1}{j-1}\lambda_s^{j-1}(1+\lambda_s)^{v+1-jp} \\
&\quad - \sum_{j=2}^{\lfloor \frac{v+1-p}{p} \rfloor+1}(-1)^{j-2}\binom{v+1-jp+j-2}{j-2}\lambda_s^{j-1}(1+\lambda_s)^{v+1-jp} \\
&= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v}{p} \rfloor}(-1)^{j-1}\binom{v-jp+j-1}{j-1}\lambda_s^{j-1}(1+\lambda_s)^{v+1-jp} \\
&\quad + \sum_{j=2}^{\lfloor \frac{v+1-p}{p} \rfloor+1}(-1)^{j-1}\binom{v+1-jp+j-2}{j-2}\lambda_s^{j-1}(1+\lambda_s)^{v+1-jp} \qquad (5.47)
\end{aligned}
$$

The second equation comes from the assumption that the proposition is true for $k = v$ and $k = v - p$. The fourth equality is just the third equality rewritten using a different summation index for the second summation. Suppose $v + 1$ is not a multiple of p. Then $\lfloor \frac{v}{p} \rfloor = \lfloor \frac{v+1-p}{p} \rfloor + 1 = \lfloor \frac{v+1}{p} \rfloor$.

Equation (5.47) can be written as

$$
\begin{aligned}
g(v+1) &= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v+1}{p} \rfloor} (-1)^{j-1} \left(\binom{v-jp+j-1}{j-1} + \binom{v-jp+j-1}{j-2} \right) \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp} \\
&= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v+1}{p} \rfloor} (-1)^{j-1} \binom{v+1-jp+j-1}{j-1} \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp} \\
&= \sum_{j=1}^{\lfloor \frac{v+1}{p} \rfloor} (-1)^{j-1} \binom{v+1-jp+j-1}{j-1} \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp}
\end{aligned}
$$

The second equality comes from the fact that $\binom{c-1}{k} + \binom{c-1}{k-1} = \binom{c}{k}$. This proves the proposition for the case when $v+1$ is not a multiple of p. Suppose $v+1$ is a multiple of p. Then $\lfloor \frac{v}{p} \rfloor = \lfloor \frac{v+1-p}{p} \rfloor = \lfloor \frac{v+1}{p} \rfloor - 1$. In this case (5.47) can be written as

$$
\begin{aligned}
g(v+1) &= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v+1}{p} \rfloor - 1} (-1)^{j-1} (\binom{v-jp+j-1}{j-1} + \binom{v-jp+j-1}{j-2}) \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp} \\
&\quad + (-1)^{\lfloor \frac{v+1}{p} \rfloor - 1} \binom{v+1-\lfloor \frac{v+1}{p} \rfloor p + \lfloor \frac{v+1}{p} \rfloor - 2}{\lfloor \frac{v+1}{p} \rfloor - 2} \lambda_s^{\lfloor \frac{v+1}{p} \rfloor - 1} (1+\lambda_s)^{v+1-\lfloor \frac{v+1}{p} \rfloor p} \\
&= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v+1}{p} \rfloor - 1} (-1)^{j-1} \binom{v+1-jp+j-1}{j-1} \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp} \\
&\quad + (-1)^{\lfloor \frac{v+1}{p} \rfloor - 1} \binom{\lfloor \frac{v+1}{p} \rfloor - 2}{\lfloor \frac{v+1}{p} \rfloor - 2} \lambda_s^{\lfloor \frac{v+1}{p} \rfloor - 1} \\
&= (1+\lambda_s)^{v+1-p} + \sum_{j=2}^{\lfloor \frac{v+1}{p} \rfloor - 1} (-1)^{j-1} \binom{v+1-jp+j-1}{j-1} \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp} \\
&\quad + (-1)^{\lfloor \frac{v+1}{p} \rfloor - 1} \lambda_s^{\lfloor \frac{v+1}{p} \rfloor - 1} \\
&= \sum_{j=1}^{\lfloor \frac{v+1}{p} \rfloor} (-1)^{j-1} \binom{v+1-jp+j-1}{j-1} \lambda_s^{j-1} (1+\lambda_s)^{v+1-jp}
\end{aligned}
$$

The second equality comes from the identity $\binom{c-1}{k} + \binom{c-1}{k-1} = \binom{c}{k}$ and from the fact that $v+1 = \lfloor \frac{v+1}{p} \rfloor p$, since $v+1$ is a multiple of p. This completes the proof of the proposition.

Starting with the Bellman equation for $w = 0$ and rearranging like (5.44) and defining $f(l) = (1 + \lambda_s)f(l-1) - \lambda_s f(l-p)$, we have,

$$\begin{aligned}\bar{h}(0) - \bar{h}(0) &= J - f(l) \\ J &= f(l)\end{aligned} \tag{5.48}$$

Using (5.48) and proposition (5.3.1), we have

$$J = g(l)\lambda_s(r - pJ)$$

Gathering terms involving J in the above equation, we obtain the following expression for J

$$\begin{aligned}J &= \frac{g(l)\lambda_s r}{1 + g(l)\lambda_s p} \\ &= \frac{g(l)\lambda r}{1 - \lambda + g(l)\lambda p}\end{aligned} \tag{5.49}$$

It is worthwhile examining the above expression. We note that λr is the expected average reward per time in the case that all arriving orders are accepted. The term $\frac{g(l)}{1-\lambda+g(l)\lambda p}$ can be interpreted as the steady state fraction of the orders that are accepted. We make this interpretation concrete as follows. Let $\theta(w,i)$ be the stationary steady state probabilities associated with state (w,i) and let $\theta(w) = \theta(w,1) + \theta(w,0)$. In words, $\theta(w)$ can be interpreted as the long term fraction of the time that the system spends in states with remaining work w. We note that $J = (\sum_{j=0}^{l-p}\theta(j))\lambda r$. Comparing this expression with 5.49, we see that $(\sum_{j=0}^{l-p}\theta(j)) = \frac{g(l)}{1-\lambda+g(l)\lambda p}$. Thus $\frac{g(l)}{1-\lambda+g(l)\lambda p}$ is the fraction of the time that the systems spends in states where an arriving order is feasible. Since an arriving order is always accepted, this is also the steady state fraction of the orders that are accepted.

We note that for a given set of parameters λ, l and p, J is a linear function of r. In other words,

$\frac{\partial J}{\partial r}$ is a constant. Also for a given set of parameters p, l and r, J is an increasing function of λ as one would expect. However $\frac{\partial J}{\partial \lambda}$ is not linear.

5.4 Static policies

As noted before, the size of the state space for OAP-B (Problem 4.5) renders the use of dynamic programming algorithms like value iteration and policy iteration impractical. In this section we consider approximations to the optimal policy for OAP-B from a special class of policies called the static policies. We define a static policy as a vector of probabilities, with one element for each order category such that every feasible order belonging to that category is accepted with the corresponding probability. In this section we consider the problem of finding the optimal policy among the class of static policies. We first develop a simple algorithm for finding the optimal static policy for OAP-B for the special case when the lead time and the processing time is the same for all the order categories. We then study the problem of finding the optimal static policy for OAP-B with the assumption that the accepted orders are scheduled using a FCFS based scheduling policy.

Static policies are appealing because of their inherent simplicity in implementation. Finding the optimal static policy has been investigated in areas like admission control to queues (see Stidham [48]), allocation of customers to servers (see Combe *et al.* [14]). In a make-to-order manufacturing context, Gallien *et al.* [24] investigate the performance of a class of static policies that accept any feasible order if and only if it belongs to a policy specific subset of the order categories. From limited computational studies, they find the performance of the optimal policy of this class competitive with more complicated heuristics. The static policies we consider are randomized policies and include the class of static policies investigated by Gallien *et al.* [24]. We now formally define the optimal static policy selection problem.

5.4.1 Optimal static policy selection problem

We define a static policy μ via a vector $\Gamma_\mu = \{\gamma_\mu^1, \ldots, \gamma_\mu^n\}$ of probabilities which specifies the probabilities with which an arriving feasible order belonging to category i is accepted for $i \in \mathcal{C}$. Consider a queuing system following the static policy μ. Every feasible order belonging to order category i is accepted with a probability γ_μ^i regardless of the history of the arrival process and control actions taken before. The expected average reward per time using a static policy μ is equal to the expected average reward per time from accepting all feasible orders with an arrival rate of $\lambda_i \gamma_\mu^i$ for the order category i. Thus choosing the optimal static policy is equivalent to choosing the optimal arrival rates $y_1, \ldots, y_n$ for the various order categories. We note that a static policy is a special randomized stationary policy. Using arguments similar to those presented in Section (5.2.3), it can be established that under any stationary randomized policy there is a single recurrent class and hence the expected average reward per time is the same starting from all states. Let $J(y_1, \ldots, y_n)$ be the expected average reward per time corresponding to accepting all feasible orders where the arrival rates are $y_1, \ldots, y_n$. We would like to choose a static policy that has the best expected average reward per time. This can be posed formally as the following optimization problem

$$\begin{aligned} max_{y_1,\ldots,y_n} \quad & J(y_1, \ldots, y_n) \\ s.t \quad & \\ 0 \leq y_i \leq \lambda_i, \quad & i \in \mathcal{C} \end{aligned} \tag{5.50}$$

5.5 Optimal static policy for the special case of equal processing times and lead times for various order categories

In this section we consider the problem of finding the optimal static policy for the class of order acceptance problems where there are multiple order categories that are differentiated only with respect to their reward. This would be the case when a firm offers essentially the same product to various customer categories for different prices. We let r_i denote the reward per order of category i. We let p denote the deterministic processing time needed for any order regardless of the order category . We also let l denote the common lead time for all order categories. Let $y_1, \ldots, y_n$ denote the arrival rates for the order categories $1, \ldots, n$ respectively. We first obtain an expression for $J(y_1, \ldots, y_n)$ which as defined in (5.4.1) is the expected average reward per time for the policy of accepting all feasible orders with arrival rates $(y_1, \ldots, y_n)$. As before, let j^t denote the category of the order arriving at time t with $j^t = 0$ indicating the non arrival of an order at that time. Let (w^t, j^t) denote the state of the system at time t. Starting from the Bellman's equation for state (w, i) for $l - 1 \geq w \geq l - p + 1$ and for $i \in \bar{C}$ and using similar arguments as in section (5.3) we see that (5.42) holds good for the class of problems under consideration in this section as well with $\bar{h}(w) = \sum_{i=0}^{n} y_i h(w, i)$. Using arguments similar to those used to derive (5.43) and (5.44) of Section 5.3, we can derive the following equations

$$\bar{h}(l - p - 1) - \bar{h}(l - p) = J - \frac{\sum_{i=1}^{n} y_i(r_i - pJ)}{1 - \sum_{i=1}^{n} y_i} \tag{5.51}$$

$$\bar{h}(w - 1) - \bar{h}(w) = J - \frac{\sum_{i=1}^{n} y_i(r_i + \bar{h}(w + p - 1) - \bar{h}(w) - J)}{1 - \sum_{i=1}^{n} y_i} \tag{5.52}$$

We now define f such that $\bar{h}(w - 1) - \bar{h}(w) = J - f(l - w)$ for $w = 1, \ldots, l - 1$. Starting with (5.52) and simplifying as in section 5.3, we have for $1 \leq w \leq l - p$,

$$f(l-w) = \frac{\sum_{i=1}^{n} y_i(r_i - pJ)}{1 - \sum_{i=1}^{n} y_i} + \frac{\sum_{i=1}^{n} y_i(\sum_{m=1}^{p-1} f(l-w-m))}{1 - \sum_{i=1}^{n} y_i} \tag{5.53}$$

For $1 \leq w \leq l - p - 1$, the above recursive equation can be simplified to,

$$f(l-w) = \left(1 + \frac{\sum_{i=1}^{n} y_i}{1 - \sum_{i=1}^{n} y_i}\right) f(l-w-1) - \left(\frac{\sum_{i=1}^{n} y_i}{1 - \sum_{i=1}^{n} y_i}\right) f(l-w-p) \tag{5.54}$$

Rewriting the last equation above using a different index, we have, for $k = p+1, \ldots, l-1$

$$f(k) = \left(1 + \frac{\sum_{i=1}^{n} y_i}{1 - \sum_{i=1}^{n} y_i}\right) f(k-1) - \left(\frac{\sum_{i=1}^{n} y_i}{1 - \sum_{i=1}^{n} y_i}\right) f(k-p) \tag{5.55}$$

Starting with the Bellman equation for $w = 0$ and rearranging like (5.52) and defining $f(l) = \left(1 + \frac{\sum_{i=1}^{n} y_i}{1-\sum_{i=1}^{n} y_i}\right) f(l-1) - \left(\frac{\sum_{i=1}^{n} y_i}{1-\sum_{i=1}^{n} y_i}\right) f(l-p)$, we have,

$$\begin{aligned} \bar{h}(0) - \bar{h}(0) &= J - f(l) \\ J &= f(l) \end{aligned} \tag{5.56}$$

Using proposition 5.3.2 with $\lambda_s = \frac{\sum_{i=1}^{n} y_i}{1-\sum_{i=1}^{n} y_i}$ and (5.56), we establish the following expression for $J(y_1, \ldots, y_n)$

$$J(y_1, \ldots, y_n) = \frac{g(l) \sum_{i=1}^{n} y_i r_i}{1 - \sum_{i=1}^{n} y_i + g(l) p \sum_{i=1}^{n} y_i} \tag{5.57}$$

5.5.1 Structure of the Optimal Static Policy

We now state a property of the optimal static policy for OAP-B with equal processing times and lead times for all order categories. We later exploit this property to formulate an algorithm for finding the optimal static policy.

Lemma 5.5.1. *There exists a set of arrival rates $y_1^*, \ldots, y_n^*$ that is an optimal solution to (5.50) such that $y_i = \lambda_i$ for $i < v$ and $y_i = 0$ for $i > v$ for some $v \in \{2, \ldots, n\}$*

Proof. The feasible set of the optimization problem (5.50) is closed and bounded. From the expression for $J(y_1, \ldots, y_n)$ given by (5.57) it is clear that $J(y_1, \ldots, y_n)$ is a continuous function for all feasible $\mathbf{y} = (y_1, \ldots, y_n)$ and hence there exists an optimal solution to (5.50). Consider an optimal solution $\bar{\mathbf{y}} = (\bar{y}_1, \ldots, \bar{y}_n)$ to (5.50) and let $J(\bar{y}_1, \ldots, \bar{y}_n) = \bar{J}$. Consider the following optimization problem.

$$\begin{aligned} max_{y_1,\ldots,y_n} \qquad & J(y_1, \ldots, y_n) \\ s.t \qquad & \\ 0 \;\le y_i \le \;& \lambda_i, i \in C \\ \sum_{i=1}^{n} y_i \;\; = \;\; & \sum_{i=1}^{n} \bar{y}_i \end{aligned} \tag{5.58}$$

We observe that for a given l and p, the function g depends only on $\sum_{i=1}^{n} y_i$. Since $\sum_{i=1}^{n} y_i$ is the same for all feasible solutions $(y_1, \ldots, y_n)$ of (5.58) it is clear from (5.57) that the objective function of (5.58) is linear. Assume without loss of generality that $r_1 \ge r_2 \ge \cdots \ge r_n$. Given the nature of the constraints, it is clear that there is an optimal solution $\mathbf{y}^* = (y_1^*, \ldots, y_n^*)$ to (5.58) that has the form $y_i^* = \lambda_i$, for $i = 1, \ldots, v-1$, $y_v^* = \sum_{i=1}^{n} \bar{y}_i - \sum_{i=1}^{v-1} \lambda_i$, $y_{v+1}^* = 0$ for some $v \in \{2, \ldots, n\}$. Note that $\mathbf{y}^*$ has the form specified in the proposition and the proof of the proposition would be complete if we show that $\mathbf{y}^*$ is an optimal solution to (5.50). Let $J(y_1^*, \ldots, y_n^*) = J^*$. It is clear that any feasible solution to (5.58) is a feasible solution to (5.50). Hence $\mathbf{y}^*$ is a feasible solution to (5.50) and so $J^* \le \bar{J}$. It can be verified that $\bar{\mathbf{y}}$ is a feasible solution to (5.58) and so $J^* \ge \bar{J}$, since $\mathbf{y}^*$ is an optimal solution to (5.58). Hence $J^* = \bar{J}$ and so $\mathbf{y}^*$ is an optimal solution to (5.50).

5.5.2 Algorithm for finding the optimal static policy

From the characterization of the optimal static policy in the above section, it is clear that if v in Lemma (5.5.1) is known then (5.50) reduces to a one-dimensional optimization problem. Let $\mathbf{y^v}$ be the optimal solution to (5.50) with the additional constraints $y_1 = \lambda_1, \ldots, y_{v-1} = \lambda_{v-1}$ and $y_{v+1} = y_{v+2} =, \ldots, y_n = 0$ for $v = 2, \ldots, n$. It is clear from Lemma (5.5.1) that $\underset{\mathbf{y^v} \in \{\mathbf{y^2}, \ldots, \mathbf{y^n}\}}{\operatorname{argmax}} J(\mathbf{y^v})$ is an optimal solution to (5.50). Thus finding the optimal static stationary policy is equivalent to solving n one dimensional problems. We now formally state the algorithm for finding the optimal static policy. We assume the existence of an algorithm that finds the global maximum of an one-dimensional optimization problem with a bounded feasible region.

1. $v = 2$;
2. $\bar{y} = \underset{\{0 \le y \le \lambda_v\}}{\operatorname{argmax}} J(\lambda_1, \ldots, \lambda_{v-1}, y, 0, \ldots, 0)$; $\mathbf{y^v} = (\lambda_1, \ldots, \lambda_{v-1}, \bar{y}, 0, \ldots, 0)$;
3. $v = v + 1$;
4. if $v = n$ return $\underset{\mathbf{y^i} \in \{\mathbf{y^2}, \ldots, \mathbf{y^n}\}}{\operatorname{argmax}} J(\mathbf{y^i})$ otherwise goto 2

5.6 Optimal Static Policy for OAP-B with FCFS based scheduling policy

In this section we assume a FCFS based scheduling policy for accepted orders for OAP-B (Problem 4.5) and investigate the problem of finding an optimal static policy under this assumption. We note that even if we are to find the optimal static policy under the assumption of FCFS based scheduling of accepted orders, we cannot expect the average reward from such a policy to be higher than the expected average reward for an optimal order acceptance policy based on FCFS scheduling of accepted orders. However, it is still worth investigating finding the optimal static order acceptance policy under a FCFS based scheduling policy because it is particularly simple to execute. Further, Gallien *et al.* [24] report that the computational performance of the optimal static policy for OAP-P (Problem 4.3) is comparable to more sophisticated heuristics. Gallien *et al.* [24] relied on simulations for finding the optimal static policy for OAP-P and this approach is not scalable for problems with a large number of order categories. We show in Chapter 6 that an appropriately defined problem of type OAP-B can be used to approximate the problem OAP-P and hence any static policy defined for the problem OAP-B can be used appropriately for problem OAP-P. The approach we take is to find the gradient of $J(y_1, \ldots, y_n)$ for problem (5.50) and use gradient based techniques for continuous optimization to find a local maxima in the space of static policies.

5.6.1 Gradient ascent

In this section we investigate finding the gradient of the expected average reward per time with respect to the arrival rates for OAP-B with FCFS based scheduling of accepted orders. For a given set of arrival rates $(y_1, \ldots, y_n)$ we seek, $\frac{\partial J(y_1,\ldots,y_n)}{\partial y_i}$, $i \in \mathcal{C}$. Since the arrival rates equivalently define a static policy, the gradient $\nabla J = (\frac{\partial J(y_1,\ldots,y_n)}{\partial y_1}, \ldots, \frac{\partial J(y_1,\ldots,y_n)}{\partial y_n})$ is also the policy gradient. Without loss of generality we assume that $l_1 = max_{i \in \mathcal{C}} l_i$. We define A_k to be the set of order categories which are feasible when the remaining work is k and B_k to be the set of order categories such that $k - p_i + 1 \geq 0$ for $i \in B_k$ when the remaining work is k. Using arguments similar to Section (5.2.3) it can be established that the Markov chain induced by any stationary policy has a single recurrent class and is aperiodic and hence there exist a unique set of stationary probabilities associated with any randomized stationary policy. A static policy is a randomized stationary policy and hence has a unique set of associated probabilities. We note that $J(y_1, \ldots, y_n)$ can be defined by the following equation

$$
\begin{aligned}
J(y_1,\ldots,y_n) &= \sum_{i=1}^{n} y_i r_i \sum_{j=0}^{l_i-p_i} \theta(j) \\
s.t \\
\theta(k) &= (1-\sum_{i\in A_{k+1}} y_i)\theta(k+1) + \sum_{i\in B_k} y_i\theta(k-p_i+1), k = l_1-2,\ldots,1 \\
\theta(0) &= (1-\sum_{i\in A_1} y_i)\theta(1) + (1-\sum_{i\in A_0} y_i)\theta(0) + \sum_{i\in B_0} y_i\theta(1-p_i) \\
\sum_{k=0}^{l_1-1} \theta(k) &= 1 \qquad (5.59)
\end{aligned}
$$

Here $\theta(k)$ is the steady state probability that the remaining work in the system is k for $k = 0,1,\ldots,l_1-1$.

The above equations show the relationship of $J(y_1,\ldots,y_n)$ to a given set of arrival rates when all feasible orders are accepted. To get the partial derivative of $J(y_1,\ldots,y_n)$ with respect to one of the arrival rates, we only have to differentiate the above set of equations with respect to that arrival rate. For example, $\frac{\partial J(y_1,\ldots,y_n)}{y_1}$ is given by

$$
\begin{aligned}
\frac{\partial J(y_1,\ldots,y_n)}{\partial y_1} &= \sum_{i=1}^{n} y_i r_i \sum_{j=0}^{l_i-p_i} \frac{\partial\theta(j)}{\partial y_1} + r_1 \sum_{j=0}^{l_1-p_1} \theta(j) \\
s.t & \\
\frac{\partial\theta(k)}{\partial y_1} &= (1-\sum_{i\in A_{k+1}} y_i)\frac{\partial\theta(k+1)}{\partial y_1} - I^1_{A_{k+1}}\theta(k+1) \\
&\quad + \sum_{i\in B_k} y_i \frac{\partial\theta(k-p_i+1)}{\partial y_1} + I^1_{B_k}\theta(k-p_1+1), k = l_1-2,\ldots,1 \\
\frac{\partial\theta(0)}{\partial y_1} &= (1-\sum_{i\in A_1} y_i)\frac{\partial\theta(1)}{\partial y_1} - I^1_{A_1}\theta(1) + (1-\sum_{i\in A_0} y_i)\frac{\partial\theta(0)}{\partial y_1} - I^1_{A_0}\theta(0) \\
&\quad + \sum_{i\in B_0} y_i \frac{\partial\theta(1-p_i)}{\partial y_1} + I^1_{B_0}\theta(1-p_1) \\
\sum_{k=0}^{l_1-1} \frac{\partial\theta(k)}{\partial y_1} &= 0 \qquad (5.60)
\end{aligned}
$$

In the above equations, $I^1_{A_k}$ is an indicator variable defined to be 1 if $1 \in A_k$ and is defined to be 0 if $1 \notin A_k$. Similarly $I^1_{B_k} = 1$ if $1 \in B_k$ and is defined to be 0 if $1 \notin B_k$. If (5.59) can be written as $\mathbf{A}\theta = \mathbf{b}$, where A, θ and b are appropriate matrices, we note from the definition of (5.60) that it can be written as $\mathbf{A}\theta = \bar{\mathbf{b}}$ where $\bar{\mathbf{b}}$ is an appropriate vector. We have shown that there exists a unique set of steady state probabilities and hence $\mathbf{A}$ is invertible. Hence there exist a unique solution to equation (5.60). Thus finding $\frac{\partial J(y_1,\ldots,y_n)}{\partial y_1}$ involves solving a set of linear equations. Hence finding the gradient of $J(y_1,\ldots,y_n)$ involves solving n linear equations each with l_1 variables.

Since the gradient of the objective function of (5.50) can be obtained for a given vector of arrival rates as shown above, we can use any of the gradient based techniques for continuous optimization which guarantee convergence to a stationary point. It should be noted that the vector of arrival rates to which any gradient based technique converges need not be an optimal solution to (5.50).

Chapter 6

Heuristics and simulations

In this chapter we describe heuristics for OAP-P (Problem 4.3) based on the results in Chapter 5. We then evaluate these heuristics using numerical simulations.

6.1 Family of discrete space problems

We note that OAP-P is a discrete time general state space problem. Our approach is to approximate this problem by a discrete time discrete state space problem, OAP-B (Problem 4.5) with appropriately chosen parameters. Given a problem of type OAP-P , we describe a family of problems $\mathcal{D}^k$, $k = 1, 2, \ldots$ of type OAP-B each of which can be considered an approximation to OAP-P at a different scale.

Let $\bar{\lambda}_i, \bar{r}_i, \bar{p}_i$ and $\bar{l}_i$ represent the arrival rate, reward, processing time and the lead time respectively for an order belonging to category $i \in \{1, \ldots, n\}$ for a given problem of type OAP-P. We let $\bar{\lambda} = \sum_{j=1}^{n} \bar{\lambda}_i$ and we define the set $\mathcal{C} = \{1, \ldots, n\}$ as before. For a given $k \in \{1, 2, \ldots\}$ we let the problem $\mathcal{D}^k$ have n order categories as well with $\lambda_i^k, r_i^k, p_i^k$ and l_i^k representing the arrival rate, reward, processing time and the lead time respectively for an order belonging to category $i \in \mathcal{C}$. We define $\lambda^k = \sum_{i=1}^{n} \lambda_i^k$. Thus λ^k is the arrival rate for the combined order arrival process for problem $\mathcal{D}^k$. We let $r_i^k = \bar{r}_i$, $p_i^k = k\bar{p}_i$, $l_i^k = k\bar{l}_i$ and $\lambda^k = 1 - e^{-\frac{\lambda}{k}}$. We also let $\lambda_i^k = \frac{\bar{\lambda}_i}{\lambda}\lambda^k$. This completely specifies all the parameters for problem $\mathcal{D}^k$.

We now present an interpretation of the arrival process and the parameters for the problem $\mathcal{D}^k$ in terms of the Poisson arrival process and the parameters of the problem (Problem 4.3). A particular realization of the order arrivals for the problem OAP-P can be represented by an infinite sequence whose elements are ordered pairs representing the time of arrival and the order category respectively of order arrivals. For example, the sequence corresponding to a specific realization of order arrivals may begin as follows $\{(1.2, 1), (2.1, 2), (2.3, 1), (3.2, 3), (3.8, 1), \dots\}$. For a given natural number k, consider a random process ζ^k constructed out of the Poisson arrival process for problem OAP-P by retaining only the first order arriving in the half open interval $[\frac{m-1}{k}, \frac{m}{k})$ for $m \in \{1, 2, \dots\}$ and pushing the time of the arrival of the order to the beginning of the interval. For example with $k = 2$, the realization of order arrivals in the process ζ^k corresponding to the order arrival sequence for the Poisson process given above begins as follows $\{(1, 1), (2, 2), (3, 3), (3.5, 1)\}$. By construction ζ^k is a discrete time process and the probability of an order arrival at times $\frac{m-1}{k}$ for $m \in \{1, 2, \dots\}$ is equal to the probability that at least one order arrives in an interval $[\frac{m-1}{k}, \frac{m}{k})$ in the original process and is equal to $1 - e^{-\frac{\lambda}{k}}$. Given that an order has arrived at time $\frac{m-1}{k}$ in the ζ^k process, the probability that the order belongs to category $i \in C$ is $\frac{\bar{\lambda}_i}{\lambda}$. Further, the probability of an order arrival at time $\frac{m-1}{k}$ is independent of the order arrivals at other times and hence ζ^k is a Bernoulli process. From the definition of the parameters for problem $\mathcal{D}^k$, it can be seen that ζ^k is the order arrival process for the problem $\mathcal{D}^k$. The processing times and the lead times for various order categories for problem $\mathcal{D}^k$ are just the corresponding parameters for the original OAP-P expressed in time units of $\frac{1}{k}$. Hence we can interpret $\mathcal{D}^k$ as a discrete space approximation to the original problem at time scale $\frac{1}{k}$. It can be seen that the process ζ^k converges to the original Poisson process as $k \to \infty$, and hence it can be expected that $\mathcal{D}^k$ approximates the original problem more accurately with increasing k.

6.2 Heuristics based on solution to $\mathcal{D}^k$

In this section we describe two heuristics, which are based on a complete solution to the OAP-B, $\mathcal{D}^k$ using usual dynamic programming methods under the additional assumption of FCFS based scheduling of accepted orders. The state at time t for problem $\mathcal{D}^k$ with FCFS based scheduling of accepted orders is (w^t, j^t) (see Section 5.2) where w^t is the remaining work at time t and j^t is the category of the order arrival at time t. The remaining work for problem $\mathcal{D}^k$ takes a value between 0 and $\bar{l}k - 1$ where $\bar{l} = \max_{i \in C} \bar{l}_i$ and there are $n + 1$ states for a particular value of remaining work. Thus, for a given k, the size of the state space for the problem $\mathcal{D}^k$ under the assumption of FCFS based scheduling of accepted orders is $(n + 1)(\bar{l}k)$. In both the heuristics a value $\hat{k}$ is chosen so that the resulting problem $\mathcal{D}^{\hat{k}}$ can be completely solved under the assumption of FCFS based scheduling of accepted orders and an optimal differential value function for the associated bellman equation is obtained. Let $h^{\hat{k}} : \mathcal{S}^{\hat{k}} \mapsto \Re$ denote an optimal differential value function for the problem $\mathcal{D}^{\hat{k}}$ under the assumption of FCFS based scheduling of accepted orders. and let $\mathcal{S}^k$ denote the state space of problem $\mathcal{D}^k$ with FCFS based scheduling of accepted orders.

We recall from Chapter 4 that the queue for OAP-P (4.3) can be described at time t, by $x^q(t) = \{(u_1^t, v_1^t), \ldots, (u_{z(t)}^t, v_{z(t)}^t)\}$ where there are $z(t)$ orders in the queue at time t and for order $a = 1, \ldots, z(t)$, (u_a^t, v_a^t) represent the remaining processing time for the a^{th} order and the time left before the a^{th} order is due respectively. We assume in this section that (u_a^t, v_a^t) are expressed in units of $\frac{1}{k}$. The state of the system at time t is given by $x(t) = \{(x^q(t), j(t))\}$, where $j(t)$ is the category of the order arriving at time t. We also recall that $\mathcal{X}^q$ is the set of all states that the queue can be in and $\mathcal{X}$ is the state space for the system. We let $w(x^q(t)) = \sum_{a=1}^{z(t)} u_a^t$ represent the remaining work for a given state $x^q(t)$ of the queue.

We now describe an order acceptance policy for the original problem, for a given stationary order scheduling policy ϕ using the optimal differential value function $h^{\hat{k}}$. For $i = 0, \ldots, n$, we define the function $\bar{h}^{\hat{k}}$ as follows

$$\bar{h}^{\hat{k}}(w,i) = h^{\hat{k}}(w,i) \quad , w = 0, \ldots, \bar{l}\hat{k} - 1$$
$$\bar{h}^{\hat{k}}(\bar{l}\hat{k},i) = h^{\hat{k}}(\bar{l}\hat{k} - 1, i)$$

We let the order acceptance policy $\mu^{\phi}_{\hat{k}}$ be defined as

$$\mu^{\phi}_{\hat{k}}(x) = 1, \quad \text{if} \quad \bar{r}_i + \sum_{j=0}^{n} \lambda^{\hat{k}}_j \bar{h}^{\hat{k}}(\lfloor w(x^q) + p^{\hat{k}}_i \rfloor, j) \geq \sum_{j=0}^{n} \lambda^{\hat{k}}_j \bar{h}^{\hat{k}}(\lfloor w(x^q) \rfloor, j) \quad (6.1)$$
$$\mu^{\phi}_{\hat{k}}(x) = 0, \quad \text{otherwise} \quad (6.2)$$

for states $x = (x^q, i) \in \mathcal{X}$ where accepting an order is feasible given the scheduling policy ϕ. Here, the action 1 represents accepting the arriving order and action 0 represents rejecting the arriving order. An interpretation of the above equation is that we are using the term $\sum_{j=0}^{n} \lambda^{\hat{k}}_j \bar{h}^{\hat{k}}(\lfloor w(x^q) \rfloor, j) - \sum_{j=0}^{n} \lambda^{\hat{k}}_j \bar{h}^{\hat{k}}(\lfloor w(x^q) + p^{\hat{k}}_i \rfloor, j)$ as an approximation to expected loss in profit due to the acceptance of an order of category i at state x. Thus, for every stationary scheduling policy ϕ, the optimal differential value function for the problem $\mathcal{D}^k$ with FCFS based scheduling of accepted orders can be used to generate an order acceptance policy.

6.2.1 FCFS-Threshold

The heuristic FCFS-Threshold uses order acceptance policy generated using $h^{\hat{k}}$ and equations (6.1) and (6.2) together with FCFS based scheduling policy for accepted orders. Let π represent the policy of FCFS based scheduling of accepted orders. The FCFS-Threshold order acceptance policy can be written using the notation established above as $\mu^{\pi}_{\hat{k}}$. From Lemma (5.2.2) and the equations (6.1) and (6.2), it can be seen that the FCFS-Threshold heuristic has order category dependent thresholds, $(w_1, \ldots, w_n)$ such that it is optimal to reject an order from category i if

$w(x^q) > w_i$. Thus, an implementation of the FCFS-Threshold requires only the storage of the thresholds $(w_1, \ldots, w_n)$.

6.2.2 FCFS-ValueFunction

The heuristic FCFS-ValueFunction uses order acceptance policy generated using $h^{\hat{k}}$ and equations (6.1) and (6.2) together with EDD based scheduling policy for accepted orders. Let ψ denote the policy of EDD based scheduling of accepted orders. The FCFS-ValueFunction order acceptance heuristic is given by $\mu^{\psi}_{\hat{k}}$. From Lemma (5.2.2) and the equations (6.1) and (6.2), it can be seen that the FCFS-ValueFunction heuristic has order category dependent thresholds, $(w_1, \ldots, w_n)$ such that it is optimal to reject an order from category i if $w(x^q) > w_i$. Thus the implementation of FCFS-ValueFunction also requires only the storage of the thresholds $(w_1, \ldots, w_n)$. However, these thresholds could be different from the thresholds corresponding to the FCFS-Threshold heuristic.

Since the heuristics FCFS-Threshold and FCFS-ValueFunction differ only in the policy used for scheduling accepted orders, FCFS-ValueFunction can be expected to perform better than FCFS-Threshold. However, it may be noted that the performance of FCFS-Threshold remains the same in the presence of costs for resuming orders where as the performance of FCFS-ValueFunction would decrease in the presence of costs for resuming orders. Hence it is interesting to numerically compute the performance loss due to FCFS scheduling of accepted orders.

6.3 Static Policies

6.3.1 FCFS-Static

FCFS-Static heuristic constructs an approximation to the optimal policy from the class of static policies. We choose a $\hat{k}$ such that it is possible to numerically solve for the set of stationary probabilities associated with any static policy for the problem $\mathcal{D}^{\hat{k}}$ under the assumption of FCFS based Scheduling of accepted orders. As mentioned in Section (5.6.1), we can follow a gradient ascent approach to obtain a local maxima in the space of static policies. FCFS-Static uses the static policy corresponding to the local maxima as the order acceptance policy and FCFS based scheduling of accepted orders.

6.3.2 EDD-Static

The EDD-Static heuristic uses the same stationary order acceptance probabilities as FCFS-Static but uses a EDD-based scheduling policy for accepted orders.

6.4 Numerical experiments

In this section, we describe the numerical experiments done to evaluate the performance of the proposed heuristics. We first describe the implementation details of the heuristics in Section (6.4.1). We then describe the numerical experiments conducted to compare the performance of various heuristics in section (6.4.2). In Section (6.5), we compare the performance of FCFS-Threshold heuristic for various values of $\hat{k}$. Finally, in Section (6.6), we describe numerical experiments designed to investigate the sub-optimality of using the FCFS based scheduling policy for accepted orders.

6.4.1 Implementation details

For all the experiments we used relative value iteration algorithm [5] for obtaining $h^{\hat{k}}$. We used 800 iterations of this algorithm starting with a vector of zeros of appropriate size for all problems. The number of iterations was sufficient for all the problems investigated in this work. In general, for all the problems investigated, the performance of the heuristics developed in this chapter did not vary much for different values of $\hat{k}$. We fix $\hat{k} = 3$ in all the subsequent experiments except for those in Section (6.5) where we compare the performance of FCFS-Threshold for two different values of $\hat{k}$.

For the FCFS-Static heuristic, we used a gradient ascent algorithm for finding the local maxima in the space of static policies. Formally, let $\mathbf{y(m)} = (y_1(m), \ldots, y_n(m))$ be the vector of arrival rates after m iterations of the algorithm. Let $\nabla J(\mathbf{y(m)})$ be the gradient of the expected average reward per time and let $||.||$ represent the Euclidean norm. We note that if $||\nabla J(\mathbf{y(m)})|| = 0$, then the algorithm has reached a stationary point and it stops. Otherwise, we define the next iterate through the following equation. For $j = 1, \ldots, n$

$$
\begin{aligned}
b_j(m+1) &= \min\{y_j(m) + \frac{\alpha(m)}{||\nabla J(\mathbf{y(m)})||} \frac{\partial J(\mathbf{y(m)})}{\partial y_j(m)}, \lambda_j^{\hat{k}}\} \\
y_j(m+1) &= \max\{b_j(m+1), 0\}
\end{aligned}
$$

An interpretation for the above equation is that the algorithm takes a step of $\alpha(m)$ in the direction of the normalized gradient $\frac{\nabla J(\mathbf{y(m)})}{||\nabla J(\mathbf{y(m)})||}$ and projects the resulting point back into the feasible region. In our experiments we used step size of the form $\alpha(m) = \frac{a}{1+bm^{0.51}}$ with $a = 0.1, b = 0.1$. This choice of step size results in $\sum_{m=1}^{\infty} \alpha(m) = \infty$ and $\sum_{m=1}^{\infty} \alpha(m)^2 < \infty$. It was observed that the convergence was rapid and to the same solution regardless of the starting point $\mathbf{y(1)}$. We used 1000 iterations of the gradient ascent algorithm for all the problems.

6.4.2 Comparison of heuristics

For the simulation results presented in this section, the methodology adopted is similar to the methodology of Gallien *et al.* [24]. In this section, we define $\bar{l}_i - \bar{p}_i = \bar{s}_i$ as the slack time for an order category i and $\bar{\sigma}_i = \frac{\bar{r}_i}{\bar{p}_i}$ as its profit rate. Similar to Gallien *et al*, we define load as $\sum_{i=1}^{n} \bar{\lambda}_i \bar{p}_i$. Thus the load is an indicator of the demand for the services of a MTO manufacturing firm. A simulation consists of a warm up period of 50 order arrivals and then a data collection period of

200 order arrivals. For each simulation, the average reward per simulation for a heuristic is the cumulative reward for the heuristic for the simulation divided by the simulation time at the 200^{th} order arrival. We conduct 30 such simulation runs and the results reported are the average over these 30 simulation runs of the average reward per simulation.

We compare the performance of the heuristics described in this Chapter with the policy of accepting all feasible orders and using an EDD based scheduling policy for accepted orders. Following Gallien *et al.*, we call this heuristic as the myopic heuristic. Where appropriate, we also compare the performance of the heuristics developed in this Chapter with the Fluid heuristic developed by Gallien *et al.*

Problem 1

We present a comparison of the performance of the heuristics FCFS-Threshold, FCFS-ValueFunction, FCFS-Static, EDD-Static and the myopic policy for the problem studied by Gallien *et al.* [24]. We describe the problem again here for convenience. The processing time of an order belongs to the set $\{1, 2, \ldots, 10\}$. The slack time of an order belongs to the set $\{5, 10\}$ and the profit rate for an order belongs to the set $\{0.7, 1.0\}$. We define order categories with the triple of processing time, slack time and profit rate taking every possible combination of values from the sets $\{1, 2, \ldots, 10\}$ and $\{5, 10\}$ and $\{0.7, 1.0\}$. Thus there are 40 order categories. The arrival rates for the order categories are defined so that $P(\bar{p}_i = p) \propto \frac{1}{p^2}$, $P((\bar{s}_i = 5, \bar{\sigma}_i = 0.7)|\bar{p}_i = p) = 0.125$, $P((\bar{s}_i = 10, \bar{\sigma}_i = 0.7)|\bar{p}_i = p) = 0.375$, $P((\bar{s}_i = 5, \bar{\sigma}_i = 1)|\bar{p}_i = p) = 0.375$, $P((\bar{s}_i = 10, \bar{\sigma}_i = 1)|\bar{p}_i = p) = 0.125$ for all $p \in \{1, \ldots, 10\}$. Together with a specific value for load, the above equations define a unique set of arrival rates.

Figure (6-1) compares the performance of the heuristics based on EDD-based scheduling of accepted orders namely FCFS-ValueFunction and EDD-Static with the myopic policy for various loads. The FCFS-ValueFunction performs well over all loads. For low loads, the myopic policy is near optimal and the gain in performance by using the FCFS-ValueFunction heuristic is not much.

However, for higher loads, FCFS-ValueFunction heuristic significantly outperforms the myopic heuristic. For intermediate and high loads, the EDD-Static significantly outperforms the myopic policy.

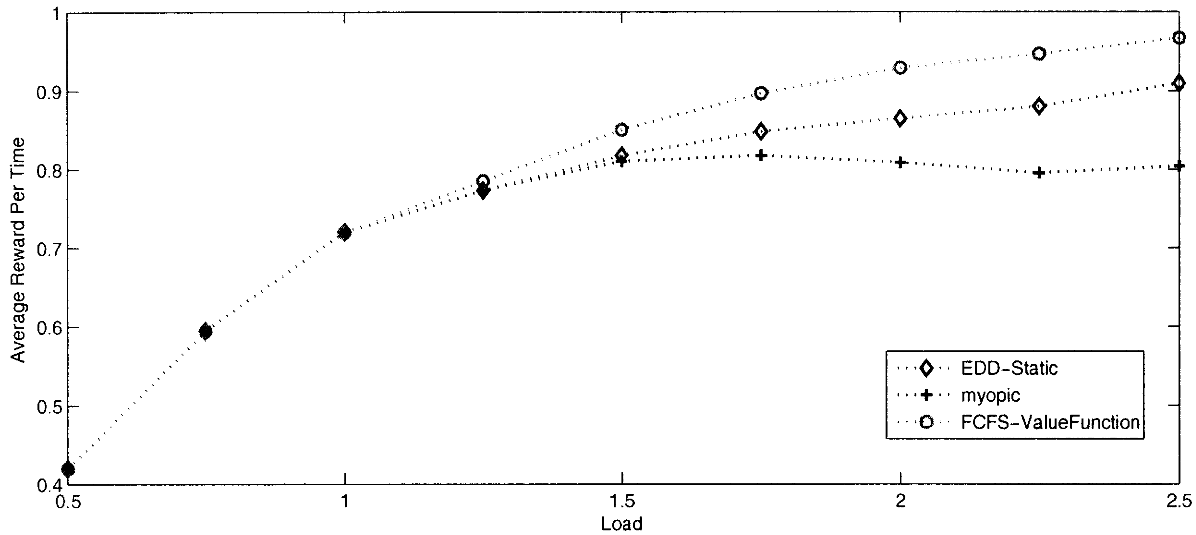

Figure 6-1: Comparison of EDD-based heuristics for various loads

Figure (6-2) compares the performance of the heuristics based on FCFS scheduling of accepted orders namely FCFS-Threshold and FCFS-Static with the myopic policy for various loads. It is interesting to note that the performance of these heuristics is very close to the performance of their counterparts based on EDD-based scheduling of accepted orders. Thus the loss of optimality by using the FCFS based scheduling policy is limited for this problem. We investigate the loss of optimality of using the FCFS based scheduling policy for accepted orders in more detail in a later section.

We now study the effect of varying the profit rates. We define order categories with the triple of processing time, slack time and profit rate taking every possible combination of values from the sets $\{1, 2, \ldots, 10\}$ and $\{5, 10\}$ and $\{\bar{\sigma}_{min}, 1.0\}$. The arrival rates for the order categories are as defined before with the load set at 1.5. Figure (6-3) compares the performance of the heuristics FCFS-ValueFunction and EDD-Static with the myopic policy for various values of $\bar{\sigma}_{min}$. The

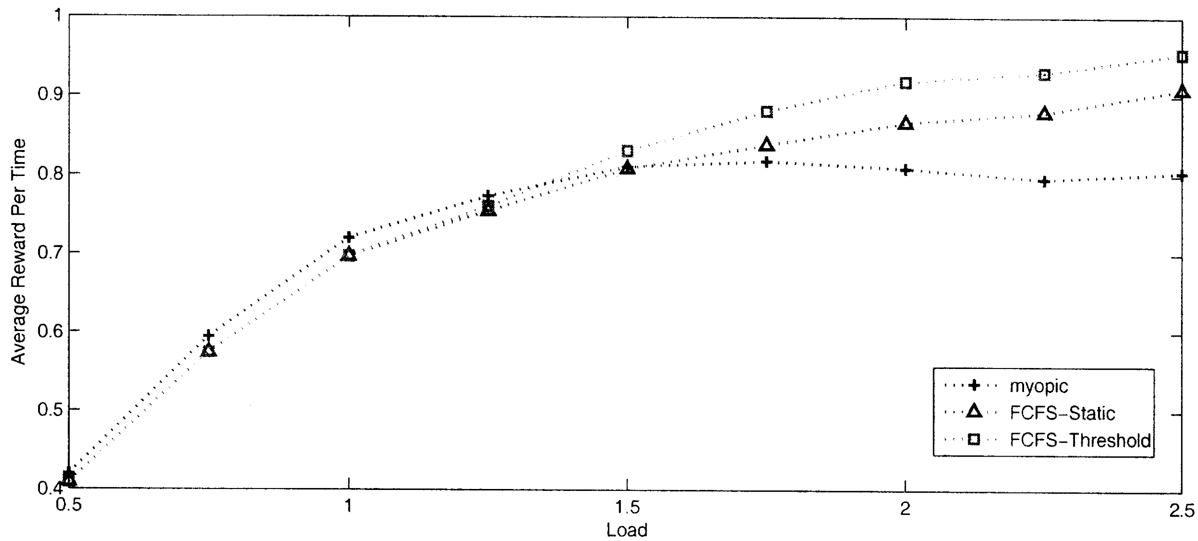

Figure 6-2: Comparison of FCFS-based heuristics for various loads

FCFS-ValueFunction heuristic significantly outperforms the myopic policy for low values of $\bar{\sigma}_{min}$ while its performance is the same as that of the myopic policy for higher values of $\bar{\sigma}_{min}$. The EDD-Static policy also outperforms the myopic policy at low values of $\bar{\sigma}_{min}$.

Figure (6-4) compares the performance of the heuristics that use FCFS based scheduling of accepted orders namely, FCFS-Threshold and FCFS-Static with the myopic policy. Again, their performance is close to the performance of their counterparts based on EDD-based scheduling of accepted orders.

Problem 2

For the next set of experiments, we let the processing time of an order belong to the set $\{1, 2, \ldots, 8\}$. The slack time of an order belongs to the set $\{12, 16\}$. We define order categories with the processing time, slack time pair taking every possible combination of values from the sets $\{1, 2, \ldots, 8\}$ and $\{12, 16\}$. Thus there are 16 order categories. For convenience, let $\bar{p}_{max} = \max_j \bar{p}_j$ and let $\bar{p}_{min} = \min_j p_j$. For an order category i with processing time $\bar{p}_i$ and slack time $\bar{s}_i$, we fix $\bar{\sigma}_i$ as

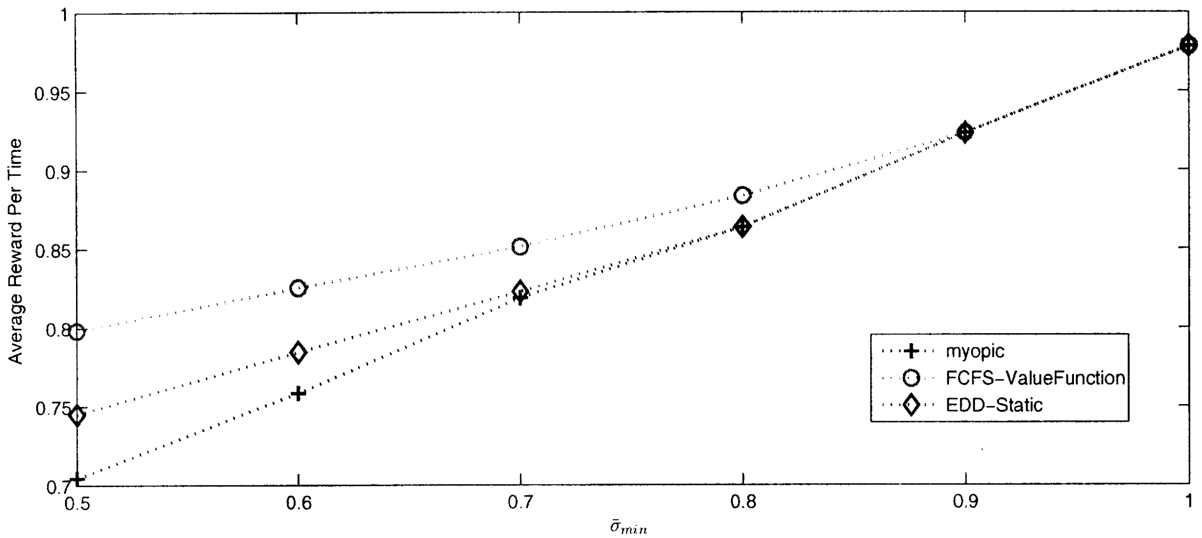

Figure 6-3: Comparison of various EDD-based heuristics for various $\bar{\sigma}_{min}$

follows,

$$\bar{\sigma}_i = 2 - \kappa \frac{\bar{p}_i - \bar{p}_{min}}{\bar{p}_{max} - \bar{p}_{min}} - \phi_i$$

The parameter κ captures the discount that the firm may offer to orders that require longer processing time. We let $\kappa = 0.5$. We assume that all customers prefer shorter lead times and ϕ is a parameter that models the discount offered to the customers for accepting higher lead times. We let $\phi_i = 0$ if $\bar{s}_i = 12$ and we let $\phi_i = 0.3$ if $\bar{s}_i = 16$.

We first study the performance of the heuristics for various loads. We let $\bar{\lambda}_i = a_1$, for $\bar{p}_i = 1, 2, 3, 4$ and $\bar{\lambda}_i = a_2$, for $\bar{p}_i = 5, 6, 7, 8$. We let $a_1 = 2a_2$. Together with the definition of load, this provides a unique set of arrival rates for a given load. We compare the performance of FCFS-Threshold, FCFS-ValueFunction, the myopic policy and the Fluid heuristic. Figure (6-5) shows the performance of the heuristics with varying load. The FCFS-ValueFunction and the FCFS-Threshold heuristics outperform the Fluid heuristic at higher loads. The performance gap between the FCFS-ValueFunction and the FCFS-Threshold heuristics is small for this problem as well.

We let $\phi_{max} = \max_i \phi_i$. We study the effect of the parameter ϕ_{max} on the performance of the heuristics. The parameter ϕ_i models the value that the customers place for a given lead time. We

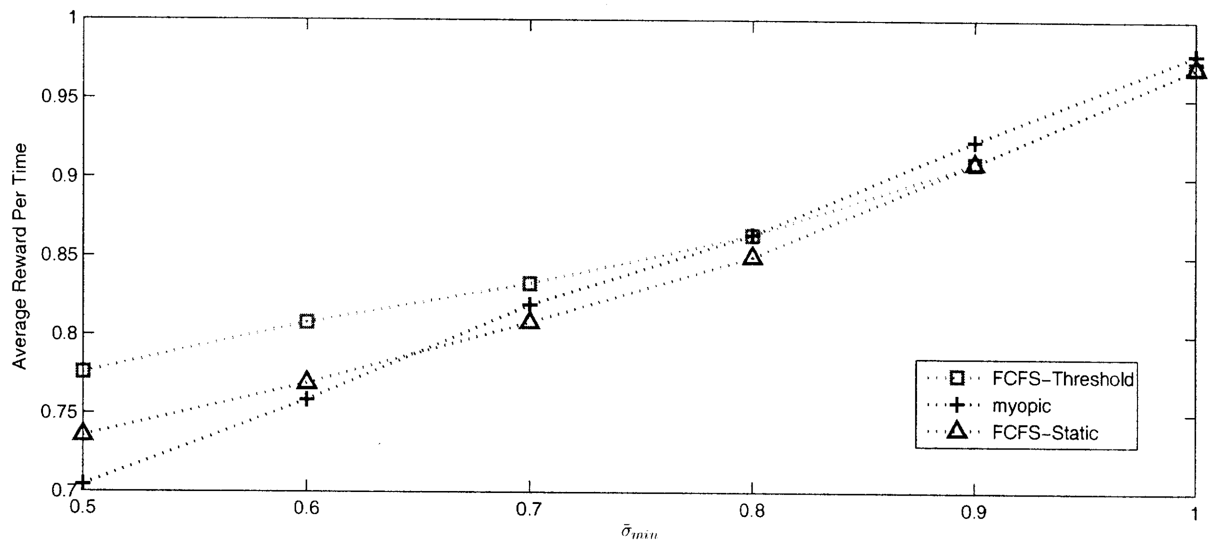

Figure 6-4: Comparison of FCFS-based heuristics for various $\bar{\sigma}_{min}$

let $\phi_i = 0$ if $\bar{s}_i = 12$ and $\phi_i = \phi_{max}$ for $\bar{s}_i = 16$. For these experiments, the load is fixed at 1.5 and the parameter $\kappa = 0$. The value of κ is set to 0 so that the variation in the profit rates of different orders is solely due to the discount offered for accepting higher lead times. All the other parameters are chosen as before. Figure (6-6) shows the performance of the heuristics for various ϕ_{max}. The performance of the FCFS-ValueFunction, FCFS-Threshold and Fluid heuristics is nearly the same for this problem. For a high value of ϕ_{max}, the performance of these heuristics is significantly better than the performance of the myopic policy.

In the last set of experiments for this problem, we study the effect of varying slack time on the performance of the heuristics. We define order categories with a processing time, slack time pair taking every possible combination of values from the sets $\{1, 2, \ldots, 8\}$ and $\{\bar{s}_{min}, 4 + \bar{s}_{min}\}$. We fix the load at 1.5 and we choose $\kappa = 0.5$ and we let $\phi_i = 0$ if $\bar{s}_i = \bar{s}_{min}$ and we let $\phi_i = 0.3$ if $\bar{s}_i = \bar{s}_{min} + 4$. Figure (6-7) presents a comparison of the heuristics for various $\bar{s}_{min}$. The FCFS-ValueFunction outperforms the Fluid heuristic for all values of $\bar{s}_{min}$. The Fluid heuristic under performs particularly at low values of $\bar{s}_{min}$. This is explained by the fact that the Fluid heuristic is based on a deterministic fluid approximation of arriving orders and this approximation is less

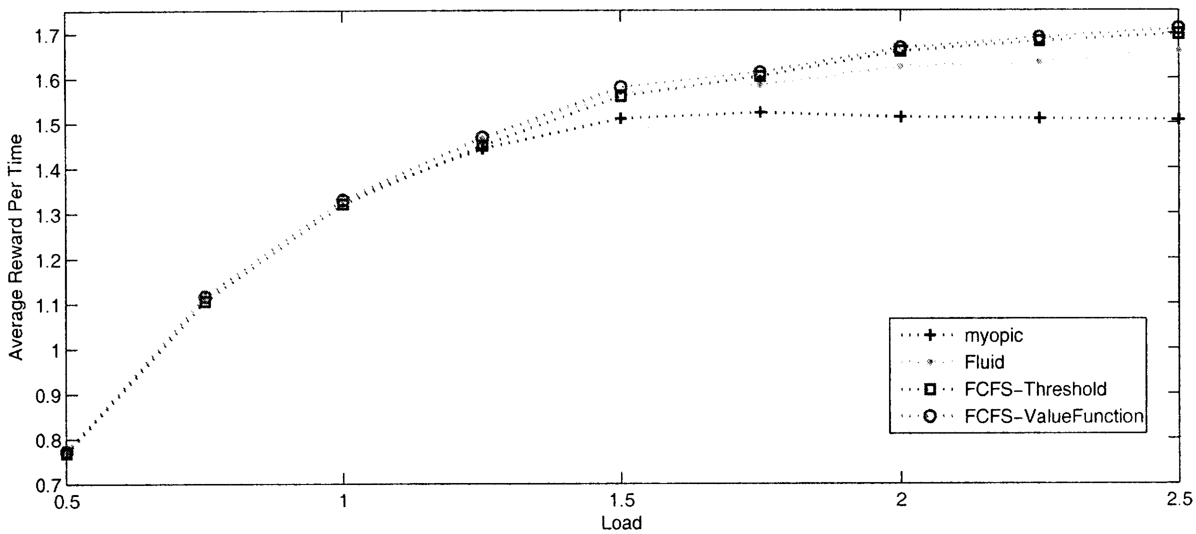

Figure 6-5: Comparison of heuristics for various loads for Problem 2

accurate for low values of $\bar{s}_{min}$.

6.5 Comparison of FCFS-ValueFunction for various values of $\hat{k}$

We reported in Section (6.4.1) that for the problems investigated in this Chapter, the performance of the heuristics did not change considerably for various values of $\hat{k}$. We consider the problem described in Section (6.4.2). Figure (6-8) shows the effect of the parameter $\hat{k}$ on the performance of the FCFS-Threshold heuristic for $\hat{k} = 1$ and $\hat{k} = 5$. It can be seen that the performance of the FCFS-Threshold heuristic is hardly distinguishable for these two values of $\hat{k}$.

6.6 Comparison of EDD and FCFS scheduling policies

We recall that FCFS-ValueFunction uses the EDD based scheduling policy for accepted orders while the FCFS-Threshold uses a FCFS based scheduling policy for accepted orders. However

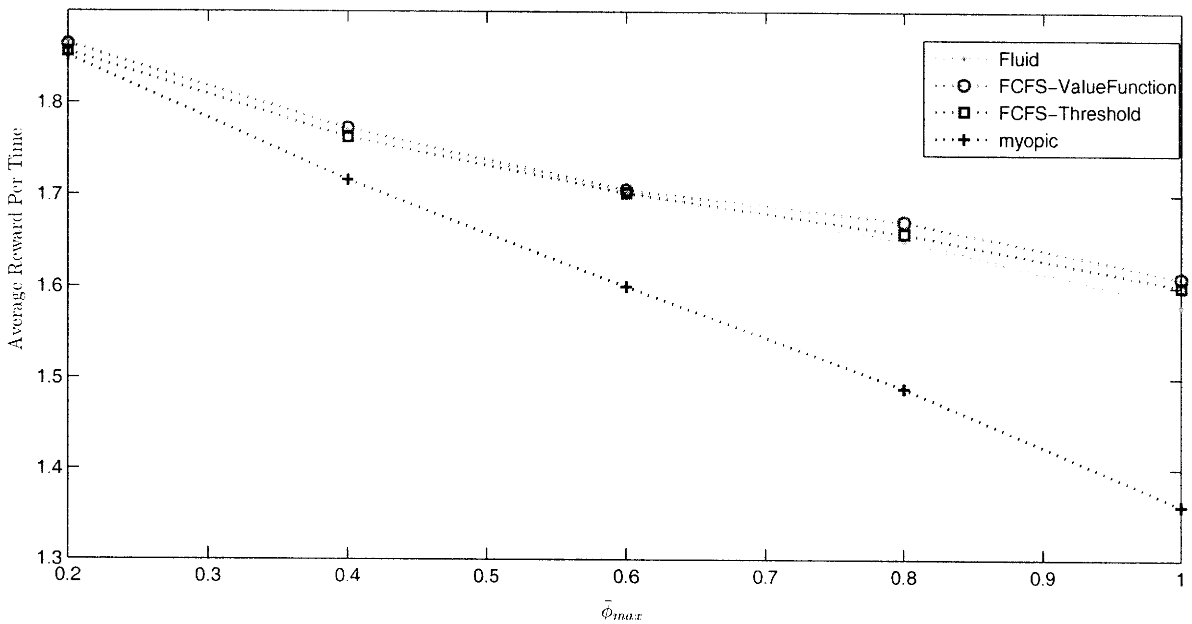

Figure 6-6: Comparison of heuristics for various $\bar{\phi}_{max}$ for Problem 2

they are based on the same function $h^{\hat{k}}$, and thus the difference in their performance is an indicator of degree of the sub-optimality of the FCFS scheduling policy. Let $\hat{a} = \frac{\max_i \bar{l}_i}{\min_i \bar{l}_i}$. If $\hat{a} = 1$, then the lead time is the same for all order categories. We note that if the lead time is the same for all order categories, the FCFS scheduling policy for the accepted orders coincides with the EDD based scheduling policy and hence the FCFS-Threshold and the FCFS-ValueFunction heuristics are the same. It can be expected that the divergence of the EDD based scheduling policy with FCFS scheduling policy increases with increasing $\hat{a}$.

For the experiments of this section, we define order categories with a processing time, lead time pair taking every possible combination of values from the sets $\{5, 6, \ldots, 14\}$ and $\{20, 20\hat{a}\}$. For an order category i with processing time $\bar{p}_i$ and lead time $\bar{l}_i$, we fix $\bar{\sigma}_i$ as follows,

$$\bar{\sigma}_i = 2 - \kappa \frac{\bar{p}_i - \bar{p}_{min}}{\bar{p}_{max} - \bar{p}_{min}} - \phi_i$$

Parameters κ and ϕ_i have a similar interpretation like in Section (6.4.2). We let $\kappa = 0.5$ and

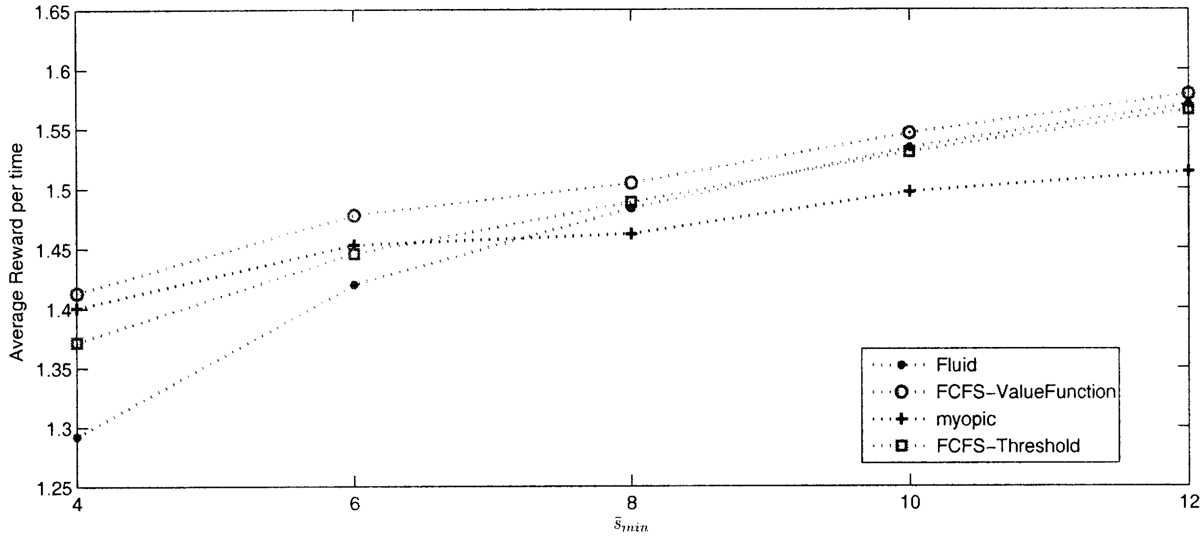

Figure 6-7: Comparison of heuristics for various $\bar{s}_{min}$ for Problem 2

$\phi_i = 0$ if $\bar{l}_i = 20$ and $\phi_i = 0.3$ if $\bar{l}_i = 20\hat{a}$. Figure (6-9) compares the performance of the heuristics FCFS-Threshold and FCFS-ValueFunction for various values of $\hat{a}$. Figure (6-10) compares the performance of the heuristics FCFS-Static and EDD-Static for various values of $\hat{a}$. The figures show the benefit of using EDD-based scheduling of accepted orders for large values of $\hat{a}$.

6.7 Summary

The FCFS-ValueFunction heuristic exhibits the best performance in the experiments performed among all the heuristics considered. The Fluid heuristic under performs the FCFS-ValueFunction heuristic when the ratio of the lead time of the orders to their processing times is low and becomes competitive with the FCFS-ValueFunction heuristic when this ratio is high. Another interesting feature of the experiments is that for a variety of situations, the performance of the heuristics that use FCFS based scheduling of accepted orders is close to the performance of their counterparts that use EDD based scheduling of accepted orders. This is useful since the performance of the FCFS scheduling policy for accepted orders does not change in the presence of resumption costs

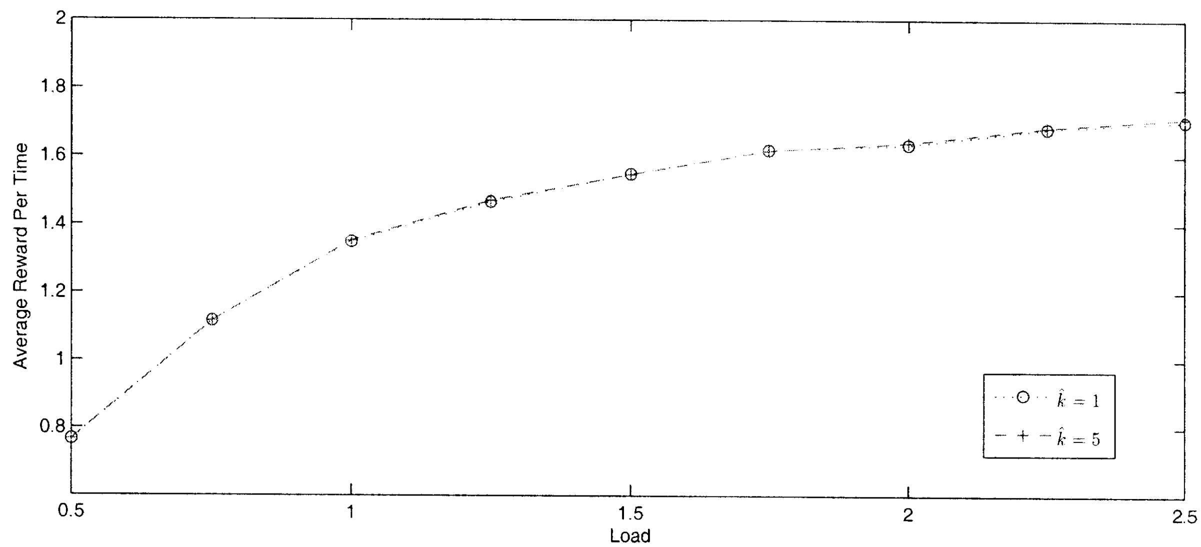

Figure 6-8: Performance of FCFS-Threshold for $\hat{k} = 1$ and $\hat{k} = 5$ for Problem 2

for orders. The performance gap between FCFS-ValueFunction and FCFS-Threshold increases with $\hat{a}$, where $\hat{a} = \frac{\max_i \bar{l}_i}{\min_i \bar{l}_i}$.

Gallien *et al.* [24] had noted that the optimal static policy performs well in comparison with the Fluid heuristic from limited computational experiments. They identified an optimal static policy using numerical simulations of all possible static policies. We adopt a broader definition of static policies and the proposed heuristics FCFS-Static and EDD-Static are computationally feasible approximations to the optimal static policy. They significantly outperform the myopic policy at intermediate and high loads. However they do not perform as well as the FCFS-ValueFunction heuristic under some conditions.

The heuristics developed in this chapter are computationally convenient to implement. For a chosen $\hat{k}$, the FCFS-ValueFunction and FCFS-Threshold heuristics require only the storage of the n order category thresholds besides the state information. To make a decision once an order arrives, the FCFS-ValueFunction and FCFS-Threshold heuristic perform just one comparison. The Fluid heuristic does not have any storage requirement besides the state information. However it involves the solution of 2 linear programming problems each time an order arrives. Hence the simulation of

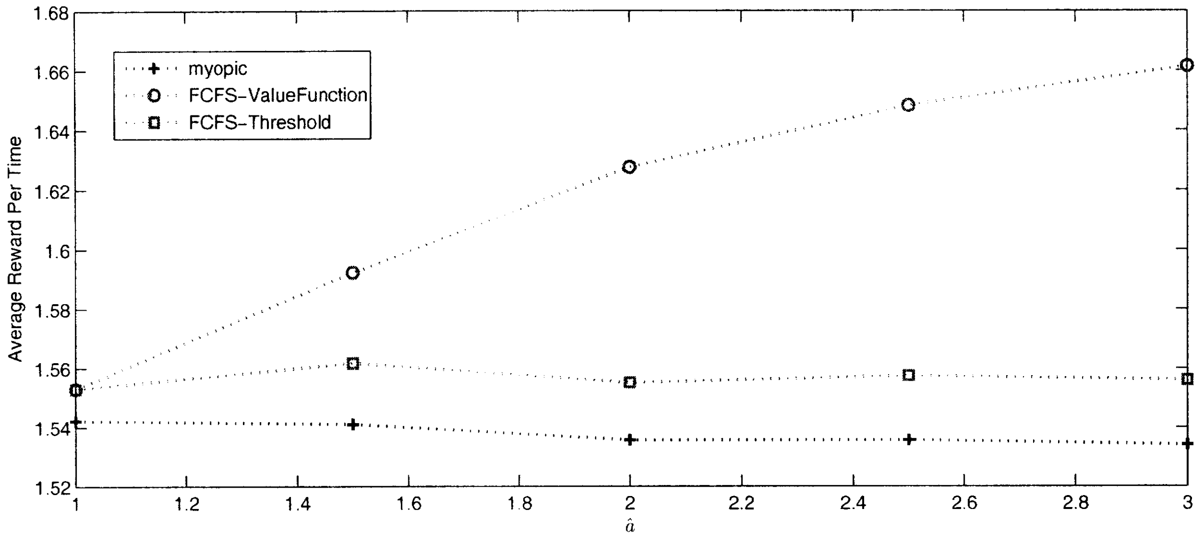

Figure 6-9: Comparison of FCFS-Threshold and FCFS-ValueFunction for various $\hat{a}$

the performance of Fluid heuristics takes much more time than the simulation of the performance of the FCFS-ValueFunction and the FCFS-Threshold heuristics. The time taken to simulate the performance of the heuristics could be an important consideration if the simulation needs to be conducted over a range of parameters.

The heuristics developed in this chapter can be extended to other related problems. For example, suppose the production facility of the MTO manufacturing firm can be modeled as a queue with m identical parallel servers and suppose an arriving order can be scheduled in any of the m servers. There exists a natural counterpart of the heuristics FCFS-Threshold, FCFS-ValueFunction, FCFS-Static and EDD-Static for these problems. Extension and evaluation of the heuristics developed in this Chapter to related problems is an interesting topic for future research.

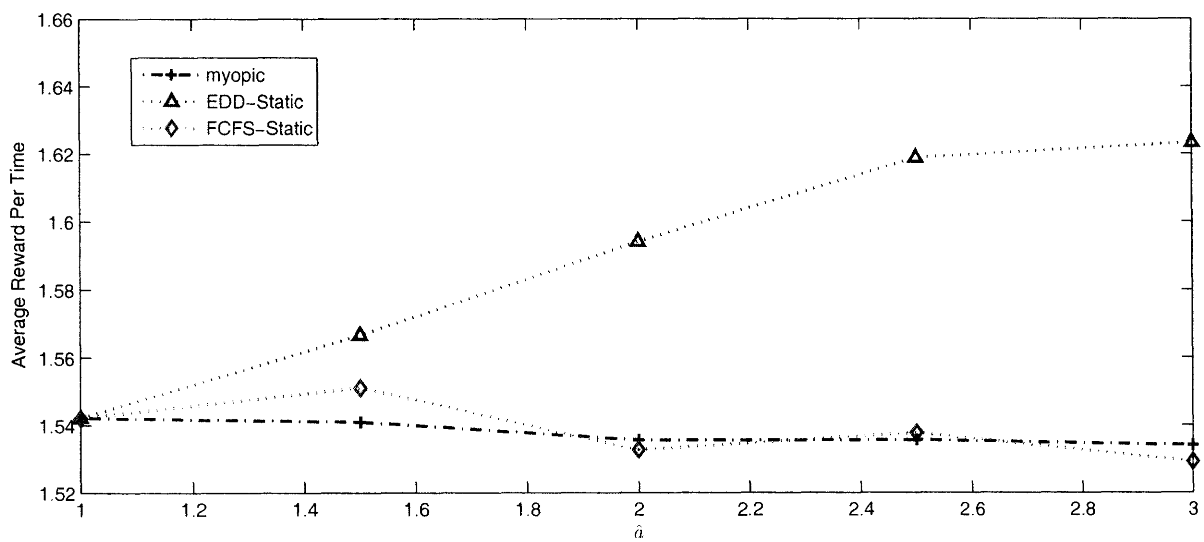

Figure 6-10: Comparison of FCFS-Static and EDD-ValueFunction for various $\hat{a}$

Chapter 7

Conclusion

Optimal control of MDPs is computationally challenging due to the curse of dimensionality. Problems involving multiple agents involve the additional challenge of dealing with limited communication among the agents. The results presented in Chapters 2 and 3 address these issues. The proposed approximation architecture directly leads to decentralized decision making while still allowing coordination. The error bound relating the choice of parameters for the linear approximation architecture to the "best error" offers guarantees on the performance of our approximation scheme. It should be noted that such guarantees are typically not available for ADP algorithms. The approximation scheme described in Chapter 2 is an extension of the ALP proposed by de Farias *et al* [16] and hence inherits its features. Specifically, through the choice of state-relevant weights and the Lyapunov function, the proposed scheme provides ways for the user of the approximation scheme to emphasize various regions of the state space as appropriate. Please see de Farias *et al* [16] for further discussion on the role of the state relevance weights and the Lyapunov functions. Further the equivalence of the linear programming problem for choosing the parameters of the linear approximation architecture to a standard class of resource allocation problems raises interesting possibilities for application of algorithms developed for resource allocation problems to the decentralized solution of the linear programming problem.

The decentralized algorithm for the class of resource allocation problems described in Chapter 3 addresses the issue of limited communication between the agents. A large class of convex optimization problems with linear constraints are equivalent to the resource allocation problem described in Chapter 3 and hence the decentralized algorithm of Chapter 3 is an optimal decentralized algorithm for a large class of convex optimization problems.

For the general state space MDP OAP-P (4.3) formulated in Chapter 4, we study a related finite state space problem MDP OAP-B (Problem 4.5) as an approximation. This technique of using a discrete state space approximation at various scales to a general state space problem can be extended to other problems involving general state spaces and Poisson arrival process.The problem OAP-B also suffers from the curse of dimensionality. We address this problem by considering FCFS scheduling of accepted orders. We characterize the optimal order acceptance policy for the problem OAP-B with FCFS Scheduling of accepted orders. A consequence of this characterization is that the heuristics FCFS-ValueFunction and FCFS-Threshold developed for OAP-P have very light storage and computational requirements. The heuristic FCFS-ValueFunction exhibits very good performance in the numerical simulations under various conditions. The performance of FCFS-Threshold is close to the performance of the FCFS-ValueFunction heuristics in many situations, suggesting that the performance loss by using FCFS scheduling of accepted orders may be minimal for the problem OAP-P. The heuristics developed are easily extendable for related problems.

The research presented in this thesis presents many possibilities for future research. We mention some of them below

- The selection of basis functions for approximation architecture has not been addressed in this thesis. This important topic is only beginning to get attention and is a rich topic for future research.

- The production facility of the MTO manufacturing firm is modeled as a single server in this work. A more realistic model would consider parallel servers where an arriving order can be scheduled in any of the servers. Extension of the results obtained in this work to such a model would be very useful.

- We formulated a model for quoting a reward, lead time pair for an arriving order and characterized the optimal policy for this problem under the assumption of FCFS scheduling of accepted orders. It would be useful to formulate heuristics similar to those presented in this thesis for this problem and evaluate their performance.

- The heuristic FCFS-Value function uses the value function derived using the assumption of FCFS scheduling of accepted orders. This can be considered as a limited form of value function approximation for the problem (4.5). It would be interesting to consider additional basis functions and use ALP techniques for deriving potentially better value function approximations for (4.5).

Bibliography

[1] K. Arrow and F.Hahn. *General Competitive Analysis.* Holden Day, San Francisco, 1971.

[2] M. Barut and V. Sridharan. Revenue management in order-driven production systems. *Decision Sciences*, 36(2):287–316, 2005.

[3] D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[4] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, second edition, 2001.

[6] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, Belmont, MA, second edition, 2001.

[7] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

[8] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability.* Athena Scientific, Belmont, MA, 2002.

[9] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, MA, 1997.

[10] O. Brun and J. Garcia. Analytical Solution of Finite Capacity M/D/1 Queues. *Journal of Applied Probability*, 37(4):1092–1098, 2000.

[11] S. Carr and I. Duenyas. Optimal admission control and sequencing in a make-to-stock/make-to-order production system. *Operations Research*, 48(5):709–720, 2000.

[12] Y-H Chang, T. Ho, and L. Kaelbling. A reinforcement learning approach to mobilized ad-hoc networks. In *International Conference on Autonomic Computing*, 2004.

[13] R. Cogill, M. Rotkowitz, B. Van Roy, and S. Lall. An approximate dynamic programming approach to decentralized control of stochastic systems. In *Proceedings of the Allerton Conference on Communication, Control, and Computing*, pages 1040–1049, 2004.

[14] M.B. Combe and O.J. Boxma. Optimization of static traffic allocation policies. *Theoretical Computer Science*, 125:17–43, 1994.

[15] D . P. de Farias. *The Linear Programming Approach to Approximate Dynamic Programming: Theory and Application.* PhD thesis, Stanford University, Palo Alto, CA, 2002.

[16] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[17] D.P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[18] D.P. de Farias and B. Van Roy. A cost-shaping LP for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations research*, 31(3):597–620, 2006.

[19] F. Defregger and H. Kuhn. Revenue Management in Manufacturing. In *Operations Research Proceedings 2003*, pages 17–22, Berlin, 2004. Springer.

[20] I. Duenyas. Single Facility Due Date Setting with Multiple Customer Classes. *Management Science*, 41(4):608–619, 1995.

[21] I. Duenyas and W. J. Hopp. Quoting Customer Lead Times. *Management Science*, 41:43–57, 1995.

[22] S. Duran. *Optimizing demand management in stochastic systems to improve flexibility and performance* . PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2007.

[23] R. Durrett. *Probability: Theory and Examples.* Duxbury Press, Belmont, CA, 1995.

[24] J. Gallien, Y.L. Tallec, and T.Shoenmeyr. A model for make-to-order revenue management. Working paper. Massachusetts Institute of Technology, Cambridge, MA, 2004.

[25] C. V. Goldman and S Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[26] A. Gravey, J. Louvion, and P. Boyer. On the Geo/D/1 and Geo/D/1/n Queues. *Performance Evalution*, 11:117–125, 1990.

[27] C. Guestrin and G. Gordon. Distributed planning in hierarchical factored MDPs. In *Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 197–206, 2002.

[28] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[29] G. M. Heal. Planning without prices. *The Review of Economic Studies*, 63:343–362, 1969.

[30] W.J. Hopp and M. R. Sturgis. A simple, robust, leadtime-quoting policy. *Manufacturing and Service Operations Management*, 3(4):321–336, 2001.

[31] A. Jalora. *Order Acceptance And Scheduling at a Make-To-Order System using Revevenue Management.* PhD thesis, Texas A&M University, 2006.

[32] J.Kurose and R.Simha. A microeconomic approach to optimal resource allocation computer systems. *IEEE Transactions on Computers*, 38, 1989.

[33] R. Kapuscinski and S. Tayur. Reliable due-date setting in a capacitated mto system with two customer classes. *Operations Research*, 55:56–74, 2007.

[34] T.S Kniker and M.H. Burman. Applications of revenue management to manufacturing. *In Third Aegean International Conference on Design and Analysis of Manufacturing Systems*, pages 299–308, 2001.

[35] H. Lakshmanan and D. P. de Farias. Decentralized resource allocation in dynamic networks of agents. *SIAM Journal on Optimization*, 19:911–940, 2008.

[36] H. Lakshmanan and D.P. De Farias. Decentralized approximate dynamic programming for dynamic networks of agents. In *American Control Conference*, 2006.

[37] P. Linwong, A. Fujii, and Y. Nemoto. Buffer-Size Approximation for the Geo/D/1/K Queue. *LNCS*, 2094:845–853, 2001.

[38] L.Xiao and S.Boyd. Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129(3):469–488, 2006.

[39] A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. on Optimization*, 12:109–138, 2001.

[40] Yu. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[41] plambeck E.L. Optimal leadtime differentiation via diffusion approximations. *Operations Research*, 52(2):213–228, 2004.

[42] M. L. Puterman. *Markov Decision Processes.* John Wiley & Sons, 1994.

[43] S. Ray and E.M. Jewkes. Customer lead time management when both demand and price are lead time sensitive. *European Journal of Operational Research*, 153(3):769–781, 2004.

[44] R.Becker, S. Zilberstein, V.Lesser, and C.V. Goldman. Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, 22, 2004.

[45] R. L Ritt and L.I. Sennott. Optimal Stationary Policies in General State Space Markov Decision Chains with Finite Action Sets. *Mathematics of Operations Research*, 17(4):901–909, 1992.

[46] P. Schwetzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.

[47] L. Servi, Y.C. Ho, and R. Suri. A class of center-free resource allocation algorithms. *Large Scale Systems*, 1:51–62, 1980.

[48] S. Stidham. Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control*, pages 705–713, 1985.

[49] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[50] T.-S. Tang and M.A. Styblinsky. Yield optimization for nondifferentiable density functions using convolution techniques. *IEEE Transactions on Computer-Aided Design*, 7(10), 1988.

[51] T.Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic approaches*. MIT press, Cambridge, Massachusetts, 1988.

[52] J.N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, MIT, 1984.

[53] J.N. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, 30(5):440–446, 1985.

[54] B. Watanapa and A. Techanitisawad. Simultaneous price and due date settings for multiple customer classes. *European Journal of Operational Research*, 166:351–368, 2005.