12-16-2016

Multi-Period Stochastic Resource Planning: Models, Algorithms and Applications

Stanislaus Simon Solomon

University of Missouri-St. Louis, ssbcd@mail.umsl.edu

Multi-Period Stochastic Resource Planning:

Models, Algorithms and Applications

Stanislaus Simon Solomon
M.S., Electrical Engineering, California State University, Sacramento, 2007
B.E., Electronics & Communications Engineering, Madras University, Chennai 2004

A Dissertation submitted to The Graduate School at the
University of Missouri – St. Louis in
partial fulfillment of the requirements for the degree
Doctor of Philosophy in Business Administration with an emphasis
in Logistics & Supply Chain Management

December 2015

<u>Dissertation Committee</u>

Haitao Li, Ph.D. (Co-chairperson)

Keith Womer, Ph.D. (Co-chairperson)

Cipriano Santos, Ph.D.

L. Douglas Smith, Ph.D.

You, Lord, keep my lamp burning;
My God turns my darkness into light.
I will say of the LORD, "He is my refuge and my fortress,
my God, in whom I trust."
- The Book of Psalms

Dedicated to my dear wife, Esha:

Thank you for your sacrificial kindness, unwavering belief, steadfast support and cheerful friendship throughout this journey.

Table of Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

Any significant accomplishment is not possible without the assistance of dedicated mentors and the sacrifice of loved ones. I am greatly indebted to my advisor Dr. Haitao Li, who introduced me to the exciting field of stochastic optimization and believed in my ability to undertake this challenging research project. It has been the best learning experience of my life. I would also like to thank the rest of my dissertation committee, Drs. Keith Womer, Cipriano Santos and Douglas Smith for their constant support and guidance. My grateful thanks to my committee for their patient mentoring throughout the past three years which helped me overcome all the challenges I have faced.

Special thanks to Dr. Cipriano Santos and Hewlett Packard Enterprises for their financial support for this dissertation. Thanks to Dr. Santos for arranging my visit to Hewlett Packard research labs at Palo Alto, CA in the fall of 2012. The visit inspired me and confirmed my desire to work on this project. A heartfelt thank you to Dr. Mary Lacity for her guidance during my Ph.D. tenure. I thank her for always being available for me and her wise counsel over the years. My grateful thanks to the Logistics and Operations Management, and the Information Systems area faculties at the University of Missouri-St. Louis (UMSL). My thanks to Karen Walsh for her friendship and help during my time at UMSL. I would also like to thank the R&D Analytics Group at Monsanto Company. The six months I spent with the group broadened my knowledge and helped me gain a deeper understanding of the role that predictive and prescriptive analytics play in solving real life problems. I would also like to thank the Management and Marketing faculty at Sam Houston State University, specifically Drs. Gerald Kohers and Pamela Zelbst, for their support towards the later stages of this dissertation.

Anyone who concentrates their efforts solely on one objective cannot help but impact those who are closest to them, which has been the case with me over the past three years. I would like to acknowledge the love and sacrifice of my wife, Esha Christie and son, Isaiah Solomon. Their dedication to my cause and patient accommodation to my needs cannot be forgotten. My wife has been a fellow traveler on this journey and I would never have completed it, if it were not up to the solidarity she provided along the way. My son has been a source of daily joy and he lifted my spirits when the going got tough.

I am eternally grateful to my parents Mr. Solomon Ebenezer (late) and Mrs. Vasanthi Solomon for their positive influence on my life. I thank my father whose dream it was for me to pursue graduate education in the U.S. He passed away before I completed my undergraduate studies but I would not be here if it wasn't up to his determined insistence. I would also like to thank my brother Abhishek Solomon for his words of encouragement and friendship over the years. I am thankful and proud of the responsible manner in which he held the fort back home while I was in grad school. My nephews Joshua and Jacob, my godson Pintoo and his sister Sasha give me joy and enrich my life.

I was privileged to enjoy the company of some wonderful colleagues during my time at UMSL. I would like to thank Srikanth Modigonda, Shaji Khan, Prasad Rudramuniyaiah, Aihua Yan, Kathy Ntalaja, William Ellegood and Dorothy Liu for their wonderful friendship and valuable help over the years. I would also like to thank Alan Dianich and Dale Schlack from Campus Crusade for Christ who mentored me during my initial years of grad school. Any attempt to measure their influence on my life will surely result in an underestimation. I would like to thank my friends Patricio La Rosa, Alicia Naveas, Nicolas Aguelakakis, Stephen Tyree, Brian Smith, Seth Waldecker, Chellamma, Aileene Sikirica, Mike Kimzey and Robin Hattori. The shared prayers, delightful meals, charming conversations and comforting encouragements lightened my load during this journey and their memories will be treasured. I would also like to thank my friend, Josh Evans, who gladly shared his expert knowledge of C++ with me. The many discussions I have had with him helped me better understand the subtle intricacies of computational algorithm development, which cannot be picked up from a textbook.

Finally, I would like to acknowledge the work of Dr. Richard Bellman. Dr. Bellman's original research in the area of multi-stage decision problems led to the development of the elegant Bellman equation and the field of dynamic programming in the 1950s. This dissertation is founded on his research and would not have been possible without his pioneering work.

ABSTRACT

This research addresses the problem of sequential decision making in the presence of uncertainty in the professional service industry. Specifically, it considers the problem of dynamically assigning resources to tasks in a stochastic environment with both the uncertainty of resource availability due to attrition, and the uncertainty of job availability due to unknown project bid outcome. This problem is motivated by the resource planning application at the Hewlett Packard (HP) Enterprises. The challenge is to provide resource planning support over a time horizon under the influence of internal resource attrition and demand uncertainty. To ensure demand is satisfied, the external contingent resources can be engaged to make up for internal resource attrition. The objective is to maximize profitability by identifying the optimal mix of internal and contingent resources and their assignments to project tasks under explicit uncertainty.

While the sequential decision problems under uncertainty can often be modeled as a Markov decision process (MDP), the classical dynamic programming (DP) method using the Bellman's equation suffers the well-known curses-of-dimensionality and only works for small size instances. To tackle the challenge of curses-of-dimensionality this research focuses on developing computationally tractable closed-loop Approximate Dynamic Programming (ADP) algorithms to obtain near-optimal solutions in reasonable computational time. Various approximation schemes are developed to approximate the cost-to-go function. A comprehensive computational experiment is conducted to investigate the performance and behavior of the ADP algorithm. The performance of ADP is also compared with that of a rolling horizon approach as a benchmark solution.

Computational results show that the optimization model and algorithm developed in this thesis are able to offer solutions with higher profitability and utilization of internal resource for companies in the professional service industry.

1. INTRODUCTION

The problem of optimally assigning resources to tasks is ubiquitous in operations management. Given a set of tasks and resources, a typical assignment problem involves finding a one-to-one matching between the tasks and resources in order to either minimize the cost of the assignments or maximize their contributions. Examples include matching jobs to machines, workers to machines and jobs to workers in a variety of contexts.

The classical assignment problem (Kuhn, 1955) is a single period problem where the availabilities of resources and tasks are known (for the period of interest) and can be assumed to be constant. The assignment problem has also been used to model various operating settings with resource limitations. Assignment problems with explicit resource constraints are known as generalized assignment problems (GAP) (Cattrysse & Van Wassenhove, 1992). GAPs have abundant applications in personnel scheduling (Kennington & Wang, 1992), project planning (Drexl, 1991) and manufacturing (Foulds & Wilson, 1999). While GAP is a well-studied approach to deal with assignment decisions under limited resources, it has two main limitations – it is both *static* and *deterministic* in nature. It is a static problem because the assignment decision is made in one period, but does not address assignment decisions involving multiple periods (e.g., weeks, months or quarters). The classical GAP is also a deterministic optimization problem because all the problem data are assumed to be constant.

Business and industrial problems require that the decision maker implement assignments to satisfy current demand while also taking into account the need to satisfy future demand. For instance, in the business consulting setting professional workers need to be assigned to projects with different durations; in the manufacturing and production environment, machines and assembly lines need to be assigned to jobs which require several periods to complete; personnel scheduling involves the assignment of skilled workers to jobs over several shifts. Assignment problems that span multiple periods have a planning horizon over which the resource allocations are planned and implemented. Multi-period problems involve making decisions, implementing them, observing new information about the problem characteristics (supply and/or demand information) as it arrives, and making further decisions using the observed information. The process is repeated again at each decision point.

Furthermore, in a multi-period assignment problem uncertainty often exists and should be explicitly addressed. For instance, in the business consulting context, project execution might be delayed due to unforeseeable circumstances, allocated resources might exit the organization, or additional work might arrive without prior notice; in manufacturing, projects can be delayed by machine failure or forecasted demand can surge or contract unexpectedly.

In the multi-period assignment setting, it is ideal to consider both the immediate performance of the current decisions and their impacts on the future. In the *sequential* decision setting, uncertainty may have impact in the form of resource and task availabilities over time. Due to the inherent presence of uncertainty it is important to dynamically adjust resource allocation over time, as more information becomes available on random variables such as resource availabilities, resource capacities and demand. Such multi-period assignment problems that address the sequential nature of decision making in the presence of uncertainty are termed *dynamic assignment problems* (Powell, 1996). The dynamic assignment problem has been applied to a wide class of applications such as dynamic routing and scheduling problems in transportation (Powell, Snow, & Cheung, 2000), the assignment of specialists and cross-trained floating workers in the production lines (Sennott, Van Oyen, & Iravani, 2006), allocating cadaveric kidneys to patient for transplantation (Su & Zenios, 2002; Su & Zenios, 2005), load matching problems in long-haul trucking (Powell, 1996) and optimizing transit times taking regulated driver working hours into consideration (Goel, 2009).

The proposed technical approach in this thesis is intended to tackle the complex multi-period assignment problems under uncertainty by: a) considering the *dynamic* (i.e., multi-period) nature of the problem via modeling the sequential characteristic of the decision process and; b) modeling the inherent *stochastic* environment involved in decision-making. In particular, we name the addressed problem the multi-period stochastic resource planning (MPSRP) problem.

The MPSRP can be informally described as follows. Consider a set of resources, with uncertain availabilities, that need to be assigned to tasks to meet stochastic demand over multiple time periods (the specific length of the planning horizon). The MPSRP aims at finding the optimal matching of resources to jobs that maximizes their contribution over the planning horizon. For example, a professional service organization may deal with thousands of employees with uncertain availabilities, and assign them weekly to a multitude of current projects and future projects which may not be won by the firm i.e., projects with uncertain win probabilities (Santos et al., 2013). Due to the inherent presence of uncertainty in this setting, it is important to dynamically and adaptively optimize resource allocation over time so that resource idleness and unplanned job reassignments are reduced. A distinctive application of the MPSRP is to optimize resource planning in the professional service industry, where heterogeneous jobs and resources are present. Moreover, the MPSRP addresses uncertainty in both demand (job availability) and supply (resource availability).

A well-known approach to deal with multi-period problems under uncertainty is to implement a rolling horizon (RH) procedure (Sethi & Sorger, 1991). In a typical RH procedure, the multi-period problem is solved at each decision point, using the realized information for the current period and forecasts for the future. The procedure implements the solution only for the current period. It makes use of forecasts of the future (that might come at a cost) and does not provide feedback between successive stages of the decision process. The RH procedure is able to update the estimates of random parameters between successive periods; however, its limitation is that it essentially relies on a deterministic solution based on point-estimates of random parameters, which does not explicitly handle uncertainty. Such a deterministic solution methodology might not provide high-quality or even feasible resource planning decision, because it may easily become infeasible due to resource or job unavailability.

A more attractive solution approach to the MPSRP is the dynamic or closed-loop policy in which resource planning decisions are made in a sequential fashion through the methodology of dynamic programming (Bellman, 1952). While the RH procedure attempts to find a deterministic solution based on point-estimates of random parameters, the closed-loop DP methodology attempts to find the optimal policy at each decision point based on the realized resource and job status, while optimizing both the immediate payoff and the expected future payoff. The closed-loop DP approach is dynamic and adaptive in nature, because it is able to observe and use the information that arrives in-between decision epochs. It also explicitly considers the impact of uncertainty in its solution paradigm.

The objective of this research is to develop a computationally tractable algorithm for obtaining a closed-loop dynamic policy for the MPSRP. The MPSRP is first described as a multi-stage sequential decision problem which enables it to be modeled as a Markov Decision Process (MDP) (Puterman, 1994). An MDP provides a modeling framework that lends itself naturally for solving sequential dynamic problems. However, a computational challenge arises in the form of the so-called curse of dimensionalities for the exact stochastic dynamic programming (SDP) procedure: (i) large number of states in the system of the MDP; (ii) large number of alternative decisions, often combinatorial in nature; and (iii) large number of scenarios of random parameters (i.e., resource and job availabilities). In order to overcome these challenges, this thesis designs and develops a computationally tractable solution procedure, called approximate dynamic programming (ADP). The essence of ADP is to approximate the exact cost-to-go function in the exact

DP. Such an approximation helps circumvent the intensive computations required when solving the exact SDP procedure via the classical Bellman recursion.

The remainder of this dissertation is organized as follows. Chapter 2 introduces the MPSRP and gives an overview of its characteristics. A literature review on the dynamic assignment problem is presented in Chapter 3. The MDP model for the MPSRP problem is developed in Chapter 4. The ADP algorithm and its approximation scheme is detailed in Chapter 5. Chapter 6 outlines the computational experiments and presents results. Conclusions and ideas for future research are discussed in Chapter 7.

2. MULTI-PERIOD STOCHASTIC RESOURCE PLANNING

2.1 Introduction

Resource planning is a critical component of efficient operations management in service settings (for e.g., healthcare, hospitality, entertainment etc.). It is important for business strategy and ensures that a service organization will be able to meet current and future demand with its available resources in a cost-effective way. Resource planning addresses multi-facet issues such as the creation of work schedules, assigning personnel to shifts, and developing cross-trained resources etc.

One such problem that is encountered frequently in the service setting is the matching of human resources to various jobs. From this point onward, the term "resources" will be used to refer to human resources i.e., skilled professionals. Resources in the service setting are characterized by specific attributes such as their education, skill set, location and work experience. They are also characterized by other intrinsic attributes such as their personal interests, willingness to work in teams, ability to handle pressure, their learning capacity and so on. For example, in the healthcare industry, nurses will tend to have differing specialties, experience levels and work shift preferences. Similarly pilots and flight attendants in the airline industry will tend to have preferences for certain routes, flight times and would have obtained aircraft specific training. Efficiently managing a diverse set of skilled professionals and matching them to their best fitting job roles is a challenge for every service organization.

2.2 The Multi-Period Stochastic Resource Planning Problem

The multi-period stochastic resource planning (MPSRP) problem is addressed in the service context. It is motivated by the resource planning challenges encountered by business consulting firms. The consulting business is characterized by firms who bid on contractual work and have resources that can be assigned to complete the work. Such firms need to match their internal workforce (IWF) to a similarly large set of diverse jobs, typically over the firm's planning horizon. The MPSRP is concerned with staffing projects that are in the firm's pipeline. A consulting firm's pipeline consists of both projects that have been won by the firm, and projects that are being bid on concurrently. Hence the planned workforce would have to staff both the realized and anticipated projects.

In the case that the firm does not have enough IWF capacity to staff the jobs in its pipeline or if the available IWF are not qualified, the jobs can be outsourced to a contingent workforce (CWF). The CWF is made of resources from an external organization who are hired temporarily to help staff jobs that cannot be satisfied with the IWF. Outsourcing jobs to the CWF ensures that the execution of projects in the pipeline proceed as planned without significant delays. The IWF have knowledge of jobs implicitly due to understanding of how the firm's business processes work. The CWF do not have the business processes know-how and may incur a learning curve while executing the jobs, depending the job specialization (Lacity, Solomon, Yan, & Willcocks, 2011). There are jobs that require "commodity" skills i.e., skills that easily substitutable (e.g., a Java developer, C++ programmer). Jobs that require non-commodity skills such as a computer scientist, statistician or operations researcher are specialized in nature and incur a learning curve effect. The pay of the IWF is greater than the CWF but the IWF are a better fit to the jobs than the CWF. These are the two type of resources considered in the MPSRP.

2.2.1 Project Decomposition

With the well-known work-breakdown-structure (WBS), a project can be decomposed into several job roles, each of which needs to be staffed by only one resource. That is, we assume that there is a 1:1 matching between a job role and a resource. For example, a project, based on its requirements, might be broken down into the following job roles: Jr. Systems Analyst I, Jr. Systems Analyst II, Sr. Java Developer, and Sr. Project Manager. Based on these job roles, the firm will assign four professionals to execute the project. This is an assumption since in practice the notion of FTE (Full Time Equivalent) is considered for project staffing. The FTE represents the resource capacity required to perform a job. For example, an FTE requirement of 0.5 for a job implies that a resource will be required 50% of its time to staff the job. FTE requirements are complicated to model because they are not equal to headcount but they need to be translated to headcount during resource allocation. Problems that consider FTE allocation will studied as future research.

2.2.2 Project Value

The value of a project (including revenue, profitability, good will, future business etc.) is shared between the jobs that make up a project.

2.2.3 Types of Project

The following are different types of projects that a typical service firm staffs:

1. Ongoing Projects:
    - Already won projects being executed currently
    - Ongoing projects account for 80%-90% of jobs at any given time
2. Project Opportunities:
    - Projects that HP are bidding on and are expected to win
    - Project opportunities account for 10%-20% of jobs at any given time
3. Unexpected Work:
    - New projects
    - Current projects extended to longer period

Projects typically last anywhere between 3 to 18 months. This is the benchmark used for generating planning horizon for the problem. It is critical for the firm to assign the appropriate resources to the jobs as haphazard or inefficient assignments can turn out to be expensive. Suboptimal assignments may lead to substandard work quality, missed deadlines, declining employee productivity and customer dissatisfaction. While assigning resources to meet demand, the firm has to ensure that none of its resources are being wasted i.e., by being left unassigned. Similarly, due to its contractual obligations the firm will need to ensure that all of the jobs are being staffed and project execution is progressing well at each epoch in the planning horizon. Another important factor to consider in this context is that the constant reassignment of a job to different resources should be avoided whenever possible. In the business consulting context we encounter a technically intensive work environment where job reassignments can be detrimental to on-the-job learning and productivity. These factors are critical and should be taken into consideration when matching the firm's resources with its demand.

2.3 Uncertainty

A key characteristic of the MPSRP is that it addresses the uncertainty encountered in the service setting. The uncertainty is in the form of IWF resource attrition and job win probability. Each IWF resource and job have a probability of being available for each decision epoch over the planning horizon.

2.3.1 Project Uncertainty

Demand uncertainty due to uncertain project bid outcomes affects resource planning because planning decisions for future projects need to made early and cannot be put off until the projects have been actually won. Moreover, ongoing projects also have an element of uncertainty in the form of their renewal. Clients may cancel their current projects or may not renew a multi-year project for subsequent years. Each project has a win probability known to the decision-maker. The jobs that belong to a project all inherit the project's win probability. When a project is won, at any particular epoch, its jobs become available for staffing and execution from that period. There can be instances where a project can be won in a specific period but execution might begin at a subsequent period. However, once the project's execution begins its job availability is observable by the decision maker at the beginning of each period of the project's duration.

2.3.2 IWF Resource Uncertainty

On the supply side, uncertainty may exist in the form of IWF resource attrition. Each IWF resource has a probability of attrition and this leads to uncertain resource availability over the planning horizon. While an IWF resource might be available at the beginning for a given period based on its probability, there is no assurance that the resource will continue to stay at the firm over the duration of that period (.i.e., month, quarter). Unlike jobs, resource status can become change during the course of a period when its assigned job might be in execution. The complete information regarding a resource's availability over a period is observable only at the end of the period. This is in contrast with the job information as job availability is completely observable at the beginning of a period. The MPSRP explicitly accounts for this difference in the observability of the random parameters. This information delay in observing resource availability in each period brings forth interesting modeling challenges and will be dealt with in greater detail in chapter 6.

Only the current status of resources and jobs are known to the decision-maker, but their future availabilities are not known. For example, a proposed bench of resources for future work might not end up fulfilling the realized demand if job and attrition estimates are conservative. Such a scenario may cause jobs to be left unstaffed and result in the need of either using the CWF, or giving-up them which might negatively affect the firm's market share and competitiveness. Similarly, if the realized demand is lower than the expected demand, there would not be enough jobs to assign to all of the planned resources. Both of these scenarios may increase costs. These additional costs are modeled as penalties to the planning decision and should be minimized. Hence the objective of the MPSRP is to maximize the net project profitability, which is the difference between the total return of staffed (assigned) projects and the total costs including both the staffing (assignment) cost and the penalty costs.

2.3.3 Supply and Demand Uncertainty

2.4 Assignment Contribution & Penalties

When a resource is matched with a job the firm realizes a contribution (or return/reward) from the assignment. Moreover there will penalties incurred when an IWF resource is unassigned (i.e., idle) and a job is reassigned. They are as follows:

1. IWF assignment contribution
2. CWF assignment contribution
3. Idle resource penalty
4. Job reassignment penalty

We discuss these in detail next.

2.4.1 IWF Assignment Contribution

The primary contribution incurred by the firm when an internal resource is assigned to fulfill a job each period. This contribution is a period specific contribution and it is incurred for each assignment made in every period of the planning horizon. The assignment contribution per period is a complex variable that includes the value obtained from staffing a job in that period, the matching score between the job and its assigned resource, and the pay of the resource for that period. The value attained from executing a job tends to be more than just the revenue – it can also include goodwill, potential for future business, and reputation enhancement.

Contribution of an assignment =

(Value of job per period) – (pay of the employee per period / matching score of IWF)

2.4.1.1 Matching Score

This component of the assignment contribution gives the decision maker an idea of the fit between each available IWF resource and each job that needs to be staffed. The matching score is calculated using the analytical hierarchy process (AHP) (Saaty, 1990) that computes weights of job attributes that reflect each job attribute's importance to executing the job. Based on the weights of the attributes we attempt to match each job to available resources. If there is a job-resource mismatch, we quantify the quality of the matching. Through this process, we develop a qualification table with both 100% matching and less than 100% matching. We take into consideration not only pay information but also the pay grades, resource location, resource expertise, and resource type (Santos et al., 2013). Moreover, the matching score can include certain psychometric factors such as personal interests, ability to work in teams, ability to handle pressure and so on. For example, a less qualified resource assigned to a project team can incur additional costs in the form of dissatisfaction of other team members. There can also be an impact on the client, in terms of likelihood of future work, if less qualified workers are assigned to their project. Such additional psychometric costs would not be incurred if the resource is fully qualified. All of these factors combined can be viewed as the “fitness” of a resource to accomplish a specific job.

2.4.2 CWF Assignment Contribution

A job can be outsourced to be performed by a CWF for two reasons: (a) there are not enough resources to staff the job; (b) the job is not valuable enough to be staffed by the IWF. The critical jobs are prioritized for the IWF resources. Non-critical jobs can be staffed either by the IWF or outsourced to CWF. The contribution incurred if a job is outsourced to CWF in a period is calculated using the value obtained from staffing a job in a period and the cost of staffing the job using a CWF.

Outsourced job contribution / period = (value of job per period) – (cost of outsourcing job to CWF per period) / matching score of CWF)

2.4.3 Idle Resource Penalty

A penalty is incurred if an IWF resource is idle (left unassigned) in a period. In practice there are no idle resources. If a resource is idle, then he/she is assigned to shadow another resource to help in their assignment. Shadow resources do not directly generate revenue but do so indirectly. When a resource is left unassigned for a specific period the firm will still need to pay them and may also incur additional training costs. This includes both the unallocated new hires and future bench (i.e., resources released from ongoing projects).

2.4.4 Job Reassignment Penalty

We also consider the issue of reassignment which is assumed to be undesirable in the current model. Under certain conditions like developing a multi-skilled workforce, job rotation is encouraged. The projects that are encountered in the consulting business are highly technical and the jobs are mostly heterogeneous in nature. It is difficult to transfer learning from one job to another even within the same project as job requirements and skills tend to vary a lot. If a job is being reassigned frequently among different resources there might be negative impacts such as the management cost of handling reassignments, the learning curve incurred by resources, possible reduction in productivity and reductions in job satisfaction. For example, in a consulting firm with several hundreds of employees working simultaneously on a lot of projects, it would take additional cost and effort by the management to keep track and to handle reassignments between periods. Moreover, project teams tend to work well when their members are familiar with each other and have established a working relationship. Frequent reassignments changes team structures which can lead to disruptions in project execution. These side effects of reassignment can lead to additional costs and they can be considered as a penalty incurred when reassignment occurs. There are 3 types of reassignments that are penalized:

1. IWF to IWF
2. IWF to CWF
3. CWF to IWF

2.5 Objective Function

The objective of MPSRP is to maximize total contributions from the assignments over the planning horizon. An optimal solution policy will assign the resources to the various jobs over time while making sure that idle IWF resources and job reassignments are minimized.

2.6 Additional Assumptions

We further assume that resources' performance on the jobs does not impact the win probability of the projects. The situation where project win probability evolve over time will studied as future research.

3. LITERATURE REVIEW

3.1 Dynamic Assignment Problems

The assignment of resources (e.g., machines, personnel, finances) to tasks has been extensively studied in operations research. Typically, resources with specific uses and characteristics must be assigned to tasks with distinctive needs. In a stochastic environment, such assignment takes place in the presence of uncertainty. The objective of dynamic assignment problems are either to minimize total costs or maximize total rewards from assigning resources to tasks in the presence of stochastic parameters such as arrival rates and availabilities. The earliest work in stochastic assignment was a resource allocation problem studied by (Ferguson & Dantzig, 1956) who consider the problem of assigning several types of aircraft to routes in the face of uncertain demand. Since that time, stochastic assignment problems have been applied in various areas included logistics, telecommunications, computer science, traffic networks and healthcare.

3.1.1 Dynamic & Stochastic Assignment Models

Much of the research on stochastic assignment problems was motivated by (Derman, Lieberman, & Ross, 1972) who introduced the so-called sequential stochastic assignment problem (hereafter referred to as the DLR model). The DLR model can be described in the following way. Consider that there are n men or workers available to be assigned to n jobs. Times required for the n jobs are independently and identically distributed. The jobs arrive in sequential fashion over time. Uncertainty is in the form of the probability of the worker being able to correctly perform a job i.e., the worker's capability or fit for the job. After a worker is assigned to a job, he is unavailable for future assignments. The problem is to assign the *n* men to the *n* jobs so as to maximize total expected reward over the planning horizon. (Derman et al., 1972) develop an optimal policy that maximizes the expected reward, which is the sum product of job values and worker capability rates over all assignments.

There have been several extensions to the DLR model over the years. Albright and Derman (1972) analyze the asymptotic behavior of the optimal policy for the DLR model. Albright (1974) extended the DLR model to consider an assignment problem that resembles a G/M/n queuing system where jobs arrive at random times and must be assigned to an individual whose processing time is exponential. Job importance, job arrival rates and processing time by each individual are assumed to be uncertain. The issue of unassigned workers is taken into account in the form of an idleness penalty cost. Kennedy (1986) deals with the case where the random demands (i.e., jobs) are not necessarily independent. Nakai (1986) develops an optimal policy for the case where states of the system are not known explicitly i.e., the problem is considered in the context of a partially observable Markov chain. The inclusion of random deadlines for jobs is considered in Righter (1989). The author deals with the case of having a single exponentially distributed random deadline for all jobs, and the case where each job has its own exponentially distributed random deadline.

David and Yechiali (1995) develop the "sequential assignment match processes" (SAMP) based on the DLR model. The SAMP model is structurally similar to the DLR wherein N candidates are waiting to be matched with M random offers that arrive sequentially and assignments are made one at a time. Each candidate and each offer is characterized by a vector of random attributes and the objective is to maximize the compatibility of the attributes from the match process. The SAMP was motivated by the donor-recipient assignment in organ transplantation. It differs from the DLR only in form of the reward structure. The reward from assigning an offer to a candidate in the DLR is a multiplicative function while the SAMP counts the matching attributes to assign reward to a match. Instead of assuming a distribution for the value of the incoming jobs, Chun and Sumichrast (2006) assume a rank based assignment where the decision maker can rank the sequentially arriving jobs from best to worst and derive an optimal assignment strategy that minimizes the sum of weighted ranks using dynamic programming. Righter (2011) extends the DLR model to consider random arrivals of workers in addition to random arrivals of jobs. It should be noted that most of these extensions are theoretical in nature and motivated much of the early research on dynamic assignment problems.

In order to introduce the impact of time on the generalized assignment problem, Kogan and Shtub (1997) developed the dynamic generalized assignment problem (DGAP). Their formulation is based on a dynamic continuous-time model which is similar to models used in optimal control theory. The model considers a set of jobs j and a set of machines (agents) m. Each machine can process a subset of the jobs and the same jobs are processed by different machines with different processing rates. A control variable, in the form of the production rate of machine m performing job j at time t, is included in the model. A job can be broken down into smaller tasks which can be processed by different machines, while making sure that each machine is assigned to only one job at a time. A flow balancing equation is introduced through the use of the inventory level of job *j* at time *t* as its tasks flows through the machines. The objective of the DGAP model is to minimize the total processing, inventory and shortage costs. Kogan, Khmelnitsky, and Ibaraki (2005) extend the DGAP by including the idea of stochastic demand and develop the stochastic, dynamic generalized assignment problem (SDGAP). The SDGAP assumes stochastic demands, and many-to-many machine-job relationships i.e., each job can be assigned to multiple agents and each agent can process multiple jobs. Every agent deals with stochastic demand in each time period and is allowed to process limited number of jobs at a time within its time-dependent capacity. The model is applied in the context of stochastic flow shop scheduling of parallel workstations and flexible manufacturing cells. Tadei and Ricciardi (1999) consider the dynamic version of the multi-level stochastic assignment problem where there is a hierarchy of supply alternatives. The information received about the supply alternatives are random and hence utility from matching the supply to demand are stochastic in nature. The authors develop a stochastic extremal process to model the evaluation of the supply and demand over time.

A different form of the stochastic generalized assignment problem was developed by Albareda-Sambola, van der Vlerk, and Fernández (2006) where uncertainty was modeled in the form of job availabilities. The demand (i.e., job availabilities) is modeled as a Bernoulli distributed parameter and the authors formulate the problem as a two stage stochastic programming model with recourse. The recourse model makes a priori assignments in the first stage and a posteriori adjustments in the second stage in order to model jobs that are either lost due to resource constraints or reassigned to other resources.

The objective is to meet all demand while minimizing assignment and penalty costs. Albareda-Sambola, Fernández, and Saldanha-da-Gama (2011) apply this two stage stochastic programming model with recourse to a facility location problem. Stage one of their model chooses the locations of the facilities, while the recourse function assigns customers to the open locations, and minimizes the penalty from unmet demand and unused locations. It should be noted that both these models differ from the MPSRP as it considers uncertainty in the availabilities of both resources and jobs, while these models consider only demand uncertainty. Furthermore, the MPSRP is a multi-period model as opposed to the two stage recourse models which consider only two successive periods at a time.

In a different perspective, (Kleywegt & Papastavrou, 1998; Papastavrou, Rajagopalan, & Kleywegt, 1996) formulate the dynamic and stochastic version of the knapsack problem (DSKP) using Markov decision processes. The DSKP deals with the issue of having limited resources (i.e., a fixed capacity knapsack) and objects to be included in the knapsack arrive randomly over time. The weights of the objects and their rewards are also random and become known upon arrival. A deadline exists after which requests cannot be accepted and the objective is to maximize expected rewards accumulated by the deadline. The secretary problem proposed by Chow, Moriguti, Robbins, and Samuels (1964) can be considered to be a specific case of the knapsack problem where each object arrives randomly one at a time and the knapsack can hold only one object. Chun, Moskowitz, and Plante (1994) consider the case where more than one object can arrive at a time (i.e., the group interview problem) and develop a backward recursive equation using dynamic programming. Using different selection criteria (e.g., minimum rank, maximum utility etc.) they develop different recursive equations and stopping rules.

A different form of the dynamic assignment problem (DAP) is modeled using game theory and stochastic user equilibrium (SUE). Lennon, McGowan, and Lin (2007) develop a game theoretic model to manage the repeated assignment of a resource between two selfish agents. Such a problem arises when the objective of the agents and of the overall system can conflict with one another. The authors consider the scenario where the resource benefits the agent with the valuable task more than the agent with a routine task. The two selfish agents are concerned only with their own reward and do not have any incentive to report their task type truthfully. The objective is to optimize system performance and the authors develop a token system such that the agents have to spend their tokens in order to bid for the resource. The two selfish agents become players in a two-person non-zero-sum game and the authors find the Nash equilibrium of the game. Similarly, Wardrop (1952) stated the first and second principles of equilibrium which is used commonly in traffic analysis models. Wardrop's first principle states that each driver, on his own, tries to minimize his travel time until the network stabilizes to an equilibrium after which no user can lower his travel time by unilaterally changing his route. Traffic flows of this kind are referred to as a "User Equilibrium" state. Stochastic models include error (assumed to be independent and identically distributed) in user perceptions which impact estimates of travel times of a route. This would result in the user choosing the optimal route based on his error-prone perceptions. Traffic flows of this kind where a user can no longer reduce his perceived travel time by unilaterally changing his current route are generally called as Stochastic User Equilibrium (SUE).

3.1.2 Applications of Dynamic Assignment Problems

While the DLR model was initially developed in the context of personnel assignment, it has been applied to several types of resource allocation problems. It has been used to the study the house selling problem (Albright, 1977), the secretary problem (Rose, 1982), organ donation problem (Su & Zenios, 2002), the job hiring problem (Ross & Wu, 2012), load sharing in computer networks (Shestak, Chong, Maciejewski, & Siegel, 2009, 2012) and the investment problem (Derman, Lieberman, & Ross, 1975). These problems are modifications of the DLR model and can be viewed as special cases of the general sequential stochastic assignment problem proposed by Derman et al. (1972).

Apart from the DLR model, other forms of dynamic assignment problems have been applied, especially in logistics and supply chain management. Dynamic fleet management and vehicle routing problems tend to be dynamic in nature and exist in a stochastic environment. Terrab and Odoni (1993) introduce the “ground hold” problem where the decision is to whether ground an aircraft before take-off based on probabilistic capacity constraints at arriving airports. Nikolaev, Jacobson, and McLay (2007) consider the problem of aviation security by developing a two stage model. Stage I deals with the purchase and install of security devices. Stage II uses the DLR model to formulate a stochastic problem that determines how to assign arriving passengers to available devices and screen them in real time. Stage I is a deterministic model, while stage II incorporates uncertainty in the form of passenger assessed threat values that results from stage I. However, both stages are solved deterministically using mixed integer programming.

McLay, Lee, and Jacobson (2010) extends the two stage aviation security model proposed by Nikolaev et al. (2007). The authors use Markov decision process to develop a sequential stochastic assignment model that sequentially assigns each passenger to a security class as they arrive.

Powell, Carvalho, Godfrey, and Simão (1995) deal with the problem of a distribution network, where supply (containers) and demand (loads) wait to be matched. Demand arises in random fashion over the network, and the challenge is to optimally move and reposition supply (the containers) to meet it. Powell and Carvalho (1997) extend the model in their previous paper by assuming a heterogeneous fleet of containers and incorporate resource substitution to handle demand while Powell and Carvalho (1998) extends the fleet management problem to include delivery time windows. Powell (1996) introduces the problem of dynamically repositioning truck drivers in anticipation of loads that arrive randomly over the distribution network. Çalışkan and Hall (2003) extend the driver repositioning problem to include the issue of drivers returning to their home terminals within a pre-specified time period. Wang, Yang, and Yang (2006) consider the problem of automated intelligent transit systems reacting dynamically to demand in order to reduce passenger wait time. The automated transit systems they consider are similar to the ones found in airports traversing a predetermined set of stops (i.e., terminals). Turner, Lee, Daskin, Homem-de-Mello, and Smilowitz (2009) develop a dynamic fleet scheduling model that aims to minimize the fleet size required to meet demand that varies over the order interval. The model allows alternate delivery times and takes into consideration customer's tolerance to early or late deliveries by modeling penalty costs.

Chen and Xu (2006) address the dynamic vehicle routing problem with hard time windows in which customer orders arrive randomly over time to be picked up within their time windows. The objective is to develop optimal vehicle routes by dispatching vehicles over time to cover all orders in minimum distance. Haghani and Jung (2005) consider the dynamic vehicle routing problem with time-dependent travel times. The problem is a DVRP with soft time windows and considers multiple vehicles with different capacities, real time service requests, and real time variations in travel times between demand nodes. Meisel, Suppa, and Mattfeld (2011) address the issue of stochastic user requests in vehicle routing which requires adjusting routes dynamically. The issue of anticipating rare events in vehicle routing (i.e., accidents) is addressed by Thomas and White (2007). The authors develop a dynamic vehicle routing problem with anticipation i.e., the model deals with the case where traffic congestion occurs from rare events. Instead of reacting (i.e., rerouting) to rare events once they occur, the model uses real time traffic information and congestion statistics to anticipate congestion (and its clearance) so that the driver can position the vehicle en-route.

Another area where dynamic assignment problems are applied is in the defense and military applications. Personnel planning and scheduling is an important optimization problem in military settings from the strategic level manpower planning (Gass, Collins, Meinhardt, Lemon, & Gillette, 1988) to operational level sailor assignment in the Navy (Holder, 2005; Li & Womer, 2009). The dynamic frequency assignment problem in military settings is an extension of the traditional frequency assignment that attempts to assign frequencies to communications throughout a battlefield deployment that avoids interference (Dupont et al., 2009). Such a model addresses the issue of dynamically assigning frequencies to new communication links as they are established, instead of changing previously assigned frequencies. The weapons-target assignment problem is experienced in combat operations where a set of targets need to be assigned to a set of weapons. The objective is to determine the number of weapons of each type to be assigned to a target that minimizes the chances of target survivability (Ahuja, Kumar, Jha, & Orlin, 2007). Powell, Bouzaiene-Ayari, Berger, Boukhtouta, and George (2011) develop a dynamic assignment model that addresses airlift operations in a military setting. Airlift operations deal with managing a fleet of aircraft to serve customer demand to move passengers or freight with time window considerations. Both demand and supply are random in nature and the objective is to maximize overall reward over the planning horizon.

3.2 Summary

The dynamic assignment problem is widely studied in a variety of application areas such as healthcare, logistics, transportation, and the military. In surveying the literature, sequential resource allocation problems that address uncertainty in both supply and demand have not been studied. While the DLR model is similar in nature to the MPSRP, there are several key differences. The most obvious distinction is that in the MPSRP, uncertainty is in the form of resource and job availabilities, and not in their arrival rates. The availabilities of employees and jobs are modeled as binary variables and they are assumed to be Bernoulli distributed parameters. The MPSRP considers uncertainty in both supply and demand, while the classic DLR model considers uncertainty only on the demand side. Righter (2011) does consider uncertainty in the arrival rates of both resources and jobs, but the approach is theoretical in nature. Moreover, since the DLR model assumes availability of the resources and the jobs, it does not address the issue of reassignment which can occur when resources or jobs become unavailable over time. The MRSRP explicitly considers the uncertainty in availability and models resource and job reassignments and their penalty costs.

The MPSRP is a unique problem that can be applied to resource allocation problems in the business, military and telecommunication settings. Most treatments of stochastic assignment problems in the literature are theoretical in nature. This is due to the computational complexity of the problem domain. To address multi-period stochastic assignment problems we need a methodology that is adaptive and handles uncertainty. There is also a need for the solution methodology to handle realistic large scale applications. Our MPSRP model and ADP solution approach contributes to the existing literature in several ways. First, we develop a model for the resource planning problem where jobs availabilities and resource attrition vary over a planning horizon. The MPSRP model developed in this dissertation can applied to resource allocation problems in various settings such as project scheduling, workforce planning and capacity planning. Second, we develop an innovative ADP solution approach that solves the MPSRP in a sequential fashion under uncertainty. Specifically, we develop an ADP training algorithm for a combinatorial optimization model such as the MPSRP. To the best of our knowledge, ours is the first attempt to design ADP training mechanisms for combinatorial optimization models under uncertainty.

4. MODEL DEVELOPMENT

We first present a deterministic integer programming (IP) formulation of the MPSRP that relies on using point estimates of the random parameters. Next, we describe a stochastic dynamic programming (SDP) model for the problem. The SDP model explicitly accounts for uncertainty in its formulation. We then discuss various extensions that can be made to the basic MPSRP problem.

4.1 Formal Problem Description

The MPSRP can be formally described as follows. Consider a services firm who needs to staff its projects with skilled resources. Each project is decomposed into the specific number of job roles required to execute it. Consider a set of jobs, J that need to be staffed with a set of resources, R over a planning horizon 1, 2, …,T. Let V_j be the value obtained from executing job $j \in J$ in each period $t \in T$ of the planning horizon. The decision-maker has two alternative resources to staff jobs i.e., internal (IWF) and contingent (CWF) resources as introduced in Chapter 2. Each IWF resource is associated with a specific salary pay per period ω_r. Each job that is outsourced to a CWF resource incurs an outsourcing cost per period ω_j. We assume a 1:1 matching between a job $j \in J$ and resource $r \in R$. That is, a job can be executed to completion by a single IWF or CWF resource. A matching score M_{rj} captures the fitness of resource $r \in R$ to each job $j \in J$, based on attributes such as job requirement, job skills, resource location, resource expertise etc. This score M_{rj} is referred to as the assignment fitness and it quantifies the qualification of an IWF resource $r \in R$ for each job $j \in J$ as a fitness score (Santos et al., 2013). The matching score N_j for CWF resources depends on the fitness of the CWF to the specific job $j \in J$ that is being outsourced. We use the fitness scores developed by (Santos et al., 2013) using a flexible matching method. The flexible matching method enables the matching resource capabilities with job requirements at less than 100 percent. When there is a perfect match between a job and a resource, the matching score equals 1. A mismatch can be represented by a fractional value between 0 and 1. The fitness scores along with the value obtained from executing a job and the resource costs are used to calculate the contribution that is gained from assigning either an IWF resource or a CWF resource to a job for execution in a period.

If a resource $r \in R$ is left unassigned in a period, the firm will incur a penalty c_r^I for keeping the resource idle for that period. Reassigning a job $j \in J$ from its currently assigned internal or contingent resource $r \in R$ to another resource is undesirable and penalized using a job reassignment penalty c_δ^j . Uncertainty is present in the problem setting in the form of resource attrition and job win probabilities. The availability of a resource $r \in R$ is treated as a Bernoulli random variable, R_{rt} with a known availability probability p_r over each period $t \in T$ of the planning horizon. The availability of a job $j \in J$ is treated as a Bernoulli random variable, J_{jt} with a known win probability p_j over each period $t \in T$ of the planning horizon, based on the project's win probability. We assume that the decision-maker knows the resource and job availability probabilities over the planning horizon. The model aims to provide an effective matching between the set of resources R and jobs J to meet demand for each period of the planning horizon. The objective of the model is to maximize the total expected contributions from staffing which includes the expected assignment contributions and the expected penalty costs.

4.2 Deterministic IP Model

The deterministic integer programming model uses point estimates (mean) of the random parameters (i.e., resource and job availabilities). In this section we present the IP formulation of the MPSRP.

4.2.1 Sets

R: Set of resources

J: Set of jobs

T: Set of time periods

4.2.2 Parameters

V_j: Value obtained from staffing & executing job j in a period

ω_r: Pay per period of IWF resource r

ω_j: Cost per period of outsourcing job j to a CWF

M_{rj}: Matching score of IWF resource r for job j

N_j: Matching score of CWF resources for job j

c_r^I: Penalty cost per period for resource r to be left idle

c_δ^j: Penalty cost per period of reassigning a job from its currently assigned IWF or CWF

IWF_{rj}: Contribution per period from assigning resource r to job j

IWF Assignment Contribution = (Value of job j per period) – (pay of the resource r per period / matching score of resource r to job j)

$$IWF_{rj} = V_j - \left(\frac{\omega_r}{M_{rj}}\right) \tag{4.1}$$

CWF_j: Contribution per period from outsourcing job j to a contingent workforce (CWF)

CWF Assignment Contribution = value of job per period – (cost of outsourcing job j to CWF per period / matching score of CWF for job j)

$$CWF_j = V_j - \left(\frac{\omega_j}{N_j}\right) \tag{4.2}$$

4.2.3 Random Parameters

The availability of a resource r is treated as a Bernoulli random variable with a known availability probability p_r (1–r's attrition rate).

$$R_{rt} = \begin{cases} 1\ if\ resource\ r\ will\ be\ available\ in\ period\ t\ with\ probability\ p_r \\ 0\ with\ probability\ 1 - p_r \end{cases}$$

The availability of a project j is treated as a Bernoulli random variable with a known win probability p_j.

$$J_{jt} = \begin{cases} 1\ if\ job\ (project) j\ will\ exist\ in\ period\ t\ with\ probability\ p_j \\ 0\ with\ probability\ (1 - p_j) \end{cases}$$

In the deterministic IP model, these random parameters are fixed at their point estimates. That is, the probabilities are fixed to the decision maker's (assumed) thresholds for resource and job availability. For example, the decision maker can fix his threshold for resource attrition to be 0.20. This implies that the decision maker will assume that resources whose probability of attrition for future periods exceeds 0.20 to be unavailable for staffing. Clearly, this method is flawed and can lead to erroneous resource planning decisions. This method can be contrasted with Monte Carlo simulation where numerous samples of the random variables are generated. The Monte Carlo samples are used instead of the point estimates in the stochastic and approximate dynamic programming procedures.

4.2.4 Decision Variables

$$x_{rjt} = \begin{cases} 1 \; if \;\; IWF \; Resource \; r \; is \; assigned \; to \; job \; j \; in \; period \; t \\ 0 \; otherwise \end{cases}$$

$$y_{jt} = \begin{cases} 1 \; if \; job \; j \; is \; staffed \; by \; CWF \; in \; period \; t \\ 0 \; otherwise \end{cases}$$

$$I_{rt} = \begin{cases} 1 \; if \; IWF \; Resource \; r \; is \; idle \; in \; period \; t \\ 0 \; otherwise \end{cases}$$

$$\delta_{jt} = \begin{cases} 1 \; if \; job \; j \; is \; reassigned \; in \; period \; t \; from \; its \; previous \; resource \; in \; period \; (t-1) \\ 0 \; otherwise \end{cases}$$

4.2.5 Objective Function

$$Maximize \sum_{r\epsilon R} \sum_{j\epsilon J} \sum_{t\epsilon T} IWF_{rj} x_{rjt} + \sum_{j\in J} \sum_{t\in T} CWF_j \, y_{jt} - \sum_{r\in R} \sum_{t\in T} c_r^I \, I_{rt} - \sum_{j\in J} \sum_{t\in T} c_\delta^j \, \delta_{jt}$$

4.2.6 Constraint Set

Constraint 1: Each job if it's available this period, can be assigned to an internal resource or outsourced to a contingent workforce

$$\sum_{r\epsilon R} x_{rjt} + y_{jt} = J_{jt} \qquad \forall\ j\epsilon J, t\epsilon T$$

Constraint 2: Each resource, if it is available this period, can be assigned to a job or be idle

$$\sum_{j\epsilon J} x_{rjt} + I_{rt} = R_{rt} \qquad \forall\ r\epsilon R, t\epsilon T$$

Constraint 3: Each available job in a period should either be assigned to the same resource as in the previous period (if that job existed in the previous period), or encounter a reassignment penalty

$(i) in\ the\ first\ period\ (i.e., t = 1)$: Constraint not applicable

$(ii) for\ all\ other\ periods\ (i.e., t > 1)$:

$$y_{jt} + x_{rjt} + \delta_{jt} \geq J_{jt} x_{rjt-1} \qquad \forall\ \ t\epsilon T, j\epsilon J, r\epsilon R$$

$x_{rjt}, I_{rt}, y_{jt}, \delta_{jt}, \gamma_{jt} \epsilon\ \{0, 1\}$

4.2.7 Limitations of the Deterministic IP Model

The deterministic IP model fixes the random parameters (i.e., resource and job availabilities) to point estimates over the planning horizon. This approach is simplistic and completely ignores the uncertain nature of the MPSRP problem. Point estimates do not capture the inherent randomness of the MPSRP decision making environment. For example, existing resources might exit the project due to attrition. . Similarly, jobs that are assumed to have been won in a specific period might end up being won in a later period, resulting in a large number of idle resources. This has motived the development of a stochastic dynamic programming model for the MPSRP to explicitly model uncertainty.

4.3 Stochastic Dynamic Programming Model

Stochastic dynamic programming (SDP) is a general approach to solving multistage sequential-decision problems that involve uncertainty. An SDP framework models decisions made in a sequential fashion.

In a typical SDP model, decisions are made in multiple stages (e.g., time periods). The solution to a SDP model requires policy-type of solution, which is a rule that prescribes a decision given the current state of the system in interest. That is, a policy will prescribe what to do (decision) under certain system of the state, while considering the impact of uncertain parameters on the future payoffs. Each stage is associated with a state of the system. In general, the state should consist of all the information needed for making a decision at the current stage. The effect of the decision made at a stage is to transform the current state to a state in the next stage. The solution procedure is designed to find an optimal policy for the overall problem - a prescription of the optimal policy decision at each stage for each of the possible states.

Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages. Therefore the optimal immediate decision depends only upon the current state and not how the system got there. This is called the Markovian property and establishes the principle of optimality for dynamic programming.

Let S_t be the current state and x_t be the current decision. $C(S_t, x_t)$ is the contribution from making decision x_t in the current state, S_t. $p(s'|S_t, x_t)$ is the probability of transitioning to next state s' if we are to take decision x_t in the current state and γ is a discount factor. The foundation of SDP is a recursive equation called as the Bellman optimality equation (Puterman, 1994) and it is written as:

$$V_t(S_t) = C(S_t, x_t) + \gamma \sum_{s' \in S} p(s'|S_t, x_t) V_{t+1}(s') \tag{4.3}$$

The bellman optimality equation states that the value of being in S_t is the sum of the immediate reward from making decision x_t in state S_t and the expected future reward from the next state if x_t is implemented in the current state. The objective is to choose x_t that maximizes the expected reward (immediate and future).

4.3.1 MDP Formulation

This section outlines the stochastic dynamic programming formulation using the terminology of Markov Decision Processes (MDPs). Stochastic dynamic programming (SDP) problems are expressed using the language of MDPs and the two terms are used interchangeably. An MDP can be used to model the SDP such that the value of state-decision pairs are estimated using the Bellman equation discussed in the previous section. The problem can be implemented as a decision tree in which all possible decisions can be enumerated for each state and the iterative optimality equation can be used to solve the tree. The MDP can be described as follows:

- Stages:

  Let T be the number of stages (i.e., number of periods in the planning horizon) and t be the label for the current stage($t = 1, \ldots, T$).

- States:

  $$S'_{rjt} = \begin{cases} 1 \; if \; job \; j \; is \; assigned \; to \; IWF \; resource \; r \; at \; the \; beginning \; of \; period \; t \\ 0 \; otherwise \end{cases}$$

  $$S''_{jt} = \begin{cases} 1 \; if \; job \; j \; is \; assigned \; to \; CWF \; at \; the \; beginning \; of \; period \; t \\ 0 \; otherwise \end{cases}$$

  The state of the system is indicated by which resource, either IWF or CWF, is currently assigned to each job at the beginning of each period. It is a record of current assignments.

- Decision Variables:

Let $x_t$be the decision variable for stage n. x_t is the set of decision variables that comprise of all decisions discussed in the deterministic IP in the section above $\{x_{rjt}, y_{rt}, z_{jt}, \delta_{jt}\}$.

- System dynamics:

The dynamics of the system at time t are given by:

$$S'_{rjt+1} = x_{rjt} \qquad \forall\, j\epsilon J, r\epsilon R \qquad (4.4)$$

$$S''_{jt+1} = y_{jt} \qquad \forall\, j\epsilon J \qquad (4.5)$$

- Decisions at each stage:

Given $S'_{rjt}, S''_{jt}, J_{jt}$ & R_{rt} the set of feasible decisions at time t are:

$\mathbb{X}(S'_{rjt}, S''_{jt}, J_{jt}, R_{rt})$ =$\{x_t$:

$$\sum_{r\epsilon R} x_{rjt} + y_{jt} = J_{jt} \qquad \forall\, j\epsilon J$$

$$\sum_{j\epsilon J} x_{rjt} + I_{rt} = R_{rt} \qquad \forall\, r\epsilon R$$

$$y_{jt} + x_{rjt} + \delta_{jt} \geq J_{jt} x_{rjt-1} \qquad \forall\, r\epsilon R, j\epsilon J$$

$$x_{rjt}, I_{rt}, y_{jt}, \delta_{jt}\, \epsilon \,\{0, 1\}$$

$\}$

We also set

$\mathbb{Y}\left(S'_{rjt}, S''_{jt},\ J_{jt}, R_{rt}\right) = \{(x_t,\ S'_{rjt+1}, S''_{jt+1})$:

$$S'_{rjt+1} = x_{rjt} \qquad \forall\ j\epsilon J, r\epsilon R, x_t \in\ X\left(S^{'}_{rjt}, S^{''}_{jt}, J_{jt}, R_{rt}\right) \tag{4.6}$$

$$S''_{jt+1} = y_{jt} \qquad \forall\ j\epsilon J, x_t \in\ X\left(S^{'}_{rjt}, S^{''}_{jt}, J_{jt}, R_{rt}\right) \tag{4.7}$$

}

The set of decisions that make up $\mathbb{X}\left(S^{'}_{rjt}, S^{''}_{jt}, J_{jt}, R_{rt}\right)$ are concerned with the current state and realizations of the random parameters in the current stage. The decisions which make up $\mathbb{Y}\left(S^{'}_{rjt}, S^{''}_{jt}, J_{jt}, R_{rt}\right)$ are concerned with how decisions made in the current stage generate the next state. That is, x_t is a feasible decision when the states of the system are S'_{rjt} and S''_{jt}, supply outcome is R_{rt}, demand outcome is J_{jt}, and applying the decision x_t on the state vectors S'_{rjt} and S''_{jt} generates the state vectors S'_{rjt+1} and S''_{jt+1} for the next time period.

- Cost-to-go function: The cost-to-go function is the total contribution of the best overall policy for the remaining stages, given that the system is in states S'_{rjt} & S''_{jt}, ready to start the next stage and selects x_t as the immediate decision. The cost-to-go function comprises of two components: the immediate contribution in the current stage and the maximum future contributions for the rest of the stages (assuming optimal decisions are taken for the rest of the stages).

$$V_t\left(S'_{rjt}, S''_{jt}\right) = C_t x_t + \gamma \sum_{S'_{rjt+1}, S''_{jt+1} \in S} p\left(S'_{rjt+1}, S''_{jt+1} \middle| S'_{rjt}, S''_{jt}, x_t\right) V_{t+1}\left(S'_{rjt+1}, S''_{jt+1}\right) \quad (4.8)$$

4.4 Challenges of Solving the SDP

While solving SDP via the Bellman recursion is guaranteed to provide optimal solutions, it suffers from two main issues:

i. The transition probabilities and rewards make up the “theoretical model” of an SDP system and obtaining them is very challenging. SDP hence suffers from the “curse of modeling” (Gosavi, 2003).

ii. SDP also suffers from the “curse of dimensionality” (Powell, 2007) that can arise in problems with a large number of states, as in the MPSRP.

For example, a system with 4 resources and 4 jobs creates a state space with 2^8 unique scenarios of resource and job availabilities. Moreover, the decisions are combinatorial in nature, which makes it computationally intractable to enumerate and visit every state-decision pair.

4.5 MPSRP Extensions

The current version of the MPSRP considers only the impact of job reassignments. The model penalizes the reassignment of a job from its currently assigned resource to another resource. This is to avoid potential frequent reassignments of a job to different resources. We focus on the reassignment of jobs, and not on the reassignment of resources. This is based on the assumption that the job reassignment penalty incorporates the impact of reassigning both the job and the resource. If the impact of reassigning resources and jobs are different, and if they need to be penalized differently, the following parameters and constraint can be added to the IP model.

Let c_δ^rrepresent the penalty of reassigning a resource. We introduce a new binary decision variable δ_{rt} to indicate whether a particular resource has been reassigning from its currently assigned job to another job in each period. The objective function will now include the resource reassignment cost term as shown below:

$$Maximize \sum_{r\epsilon R}\sum_{j\epsilon J}\sum_{t\epsilon T} IWF_{rj}x_{rjt} + \sum_{j\in J}\sum_{t\in T} CWF_j\, y_{jt} - \sum_{r\in R}\sum_{t\in T} c_r^I\, I_{rt} - \sum_{j\in J}\sum_{t\in T} c_\delta^j\, \delta_{jt} - c_\delta^r \sum_{r\in R}\sum_{t\in T} \delta_{rt}$$

An additional constraint is required to model resource reassignment.

$$(y_{rt} + x_{rjt} + \delta_{rt})J_{jt} + (y_{rt} + \sum_j x_{rjt})(1 - J_{jt}) \geq R_{rt}x_{rjt-1} \qquad \forall\, r\epsilon R, j\epsilon J, t\epsilon T$$

Resource reassignment is different from job reassignment in implementation. All reassignments of a job need to be penalized. However, a resource can be reassigned without penalty if it has executed its previously assigned job to completion. The resource is then available for a new assignment and it can be reassigned to a different job without penalty. The above constraint models this feature.

A job can be left unstaffed in a period if there are no sufficient internal resources to staff all the jobs that exist in that period. There are two different ways to handle such a scenario. The first way is to make sure that the job is completed by hiring a contingent workforce (CWF) through outsourcing (at a higher cost). If the firm ends up hiring the CWF to meet demand, then the assignment cost per period for the CWF will be:

Fitness cost of CWF to that job + CWF sunk cost for that period – Value from completing that period's component of the job

If the firm hires the CWF but doesn't need to use them in a period, then it will not incur the idle resource cost as the CWF are paid only when they are assigned to work. The second way to handle the scenario is to abandon the jobs. If that is the case, then the assignment cost will be zero (as no resources are assigned to the jobs) but the firm might incur additional business costs like contractual penalties, loss of business reputation, reduced chances of future contracts and so on.

5. ADP ALGORITHM

We have seen two different models and solution methodologies to solve the MPSRP in the previous chapter. The first one is the deterministic IP which uses point estimates of the random parameters. It is clearly a naïve approach as it solves the problem for the entire horizon in an open-loop fashion, without obtaining new information about the state of the system in between periods. The deterministic IP solution would not be useful for decision making if the actual realization of the random parameters deviates from the assumed point estimates. This is the so called flaw of averages that is common in deterministic models. The second approach is the stochastic dynamic programming model which uses the Bellman optimality equation. SDP is better than deterministic approaches as it a) explicitly accounts for uncertainty and b) solves the problem in a sequential fashion, capturing the impact of uncertainty and the decisions made on future stages.

The ADP methodology can help overcome the computational challenges faced in SDP. In order to overcome the curses of dimensionality, we use Monte-Carlo simulation to simulate sample paths for the system evolution. We also use stochastic approximation methods to estimate the value function without transition probabilities, overcoming the curse of modeling. We develop the ADP formulation in this chapter. Certain benchmarks which will be used to compare the results of the ADP methodology will also discussed.

5.1 DP Approximation Schemes

Sequential decision problems can be modeled using both continuous states & time, as well as discrete states and time. Since the MPSRP is modeled using discrete states (resource and job availabilities assumed to be Bernoulli distributed parameters) and time, our approach parallels that of reinforcement learning (RL) in artificial intelligence and approximate dynamic programming (ADP) in operations research. Both of these areas have developed approximation schemes to overcome the curses of modeling and dimensionality specified above. We will first review reinforcement learning, followed by approximate dynamic programming.

5.1.1 Reinforcement Learning

Gosavi (2003) provides a detailed analysis of RL concepts and much of the material in this section is adapted from that text. RL can be viewed as a way of implementing DP algorithms within a simulator. RL algorithms help overcome the curse of modeling as the model-free algorithms of RL do not need the transition probabilities. RL can solve MDPs without the theoretical model and can still produce high quality near optimal solutions. Similarly, RL uses function approximation methods such as neural networks, regression and interpolation which need a small number of scalars to approximate the value function of millions of states.

Note that the main tool used by RL algorithm is simulation. In fact, RL also been referred to as “simulation based dynamic programming” (Gosavi, 2003). It uses simulation to avoid calculating transition probabilities and the transition rewards are automatically calculated within the simulator. RL theory is based on two fundamental concepts which we will discuss below: a) The Q-factor and, b) Robbins-Monro algorithm

5.1.1.1 The Q-Factor

The classic value iteration algorithm used to solve MDPs calculates the "value function" of each state. RL algorithms also calculate the value function but store them in the form of Q-factors. In RL, each element of the Q-factor vector is related to a "state-action" pair. It is evident then, that if the Q-factors are known, one can find the value function of a given state. The value function associated with the optimal policy for discounted reward MDPs is defined by the Bellman optimality equation as:

$$J^*(i) = \max_{a \in A(i)} [\sum_{j=1}^{S} p(i,a,j)\,[r(i,a,j) + \lambda J^*(j)]] \tag{5.1}$$

Where,

i. $J^*(i)$ is the ith element of value function vector associated with the optimal policy.
ii. $r(i,a,j)$ is the immediate reward earned when action a is selected in state i and the system transitions to state j.
iii. $p(i,a,j)$ is the probability of transitioning to state j when action a is selected in state i.
iv. λ is a discount factor for future contributions.

For a given state-action pair, the Q-factor can be defined as:

$$Q(i,a) = \sum_{j=1}^{s} p(i,a,j) \left[r(i,a,j) + \lambda \max_{b \in A(j)} Q(j,b) \right] \quad (5.2)$$

The above equation is fundamental to RL and can be viewed as the Q-Factor version of the Bellman optimality equation for discounted reward MDPs.

5.1.1.2 Robbins-Monro Algorithm

The Robbins-Monro (RM) algorithm (Robbins & Monro, 1951) helps estimate the mean of random variable from its samples. If we denote the i^{th} independent sample of a random variable X by S^i and the expected value by E(X), then the estimated produced by $\frac{\sum_{i=1}^{n} S^i}{n}$ tends to the real value of the mean as $n \rightarrow \infty$ as a result of the law of large numbers. The RM algorithm is derived from this straight forward averaging process. If we denote the estimate of X in the n^{th} iteration, that is, after n samples have been obtained by X^n:

$$X^n = \frac{\sum_{i=1}^{n} S^i}{n} \quad (5.3)$$

After transformations the above term can be defined as:

$$X^{n+1} = (1 - \alpha^{n+1})X^n + \alpha^{n+1}S^{n+1} \qquad (\text{if } \alpha^{n+1} = \frac{1}{n+1})$$

The above equation is referred to as the RM algorithm or the RM scheme. α is called the step size or learning rate. When ($\alpha^{n+1} = \frac{1}{n+1}$) the RM algorithm is directly equivalent to averaging. Other forms of α^{n+1} can be used as long as they indirectly perform averaging. The RM scheme can be used for estimating Q-factors. It can be shown that every Q-factor can be expressed as an average of a random variable:

$$Q(i,a) = \sum_{j=1}^{s} p(i,a,j)\ [r(i,a,j) + \lambda \max_{b \in A(j)} Q(j,b)]$$

$$Q(i,a) = \mathrm{E}\,[r(i,a,j) + \lambda \max_{b \in A(j)} Q(j,b)]$$

$$Q(i,a) = \mathrm{E}\,[\mathrm{SAMPLE}] \tag{5.4}$$

Due to the difficulty in obtaining the transition probabilities, the idea is to remove the expectation operator using the RM scheme. If samples of the random variable can be generated within a simulator, we can use the RM scheme to estimate the Q-factor.

$$\mathrm{Q}^{n+1}(i,a) = (1-\alpha^{n+1})\mathrm{Q}^{n}(i,a) + \alpha^{n+1}\left[r(i,a,j) + \lambda \max_{b \in A(j)} Q(j,b)\right] \tag{5.5}$$

Such an algorithm that does not use the transition probabilities in its updating equation is called as “model-free” algorithm. For the estimation of the Q-factors to be perfect, we must obtain, theoretically, an infinite number of samples of each Q-factor i.e., each state-action pair must be tried infinite times. An effective strategy is to try each action in each state with equal probability and simulate the system in such a way so that each state-action pair is tried a large number of times. The simulator will take the system from one state to another selecting each action with equal probability in each state. The RL algorithm, which will be embedded with the simulator, will update the values of the Q-factors. The values of the Q-factors are stored in a lookup table explicitly. This is feasible only for a manageable number of state-action pairs. When we have a huge number of state-action pairs, function approximation methods can be used in which not all Q-factors are stored explicitly.

5.1.2 OR Based ADP Algorithms

The ADP methodology encountered in the OR literature is quite similar to that of reinforcement learning. The exact DP methodology uses the value iteration algorithm to visit each possible state and computes the impact of every feasible decision, in each stage of the problem. It then steps back in time and exactly computes the value function which is used to produce optimal decisions. The value function provides the expected value of each decision which is the sum of the immediate reward and the expected discounted future rewards. It is evident that value iteration is not a practical strategy for even small problem sizes due to the curse of dimensionality. Powell (2007) shows that there can be three different curses of dimensionality for certain problems:

- The state space: If the state variable $S_t = (S_{t1}, S_{t2}, \dots, S_{tI})$ has I dimensions, and if S_{tI} can take on L possible values, then we might have up to L^I different states.
- The outcome space: The random variable $W_t = (W_{t1}, W_{t2}, \dots, W_{tJ})$ might have J dimensions. If W_{tJ}can take on M outcomes, then our outcome space might take on up to M^J outcomes.
- The action space: The decision vector $X_t = (X_{t1}, X_{t2}, \dots, X_{tK})$ might have K dimensions. If X_{tK} can take on N outcomes, we might have up to N^K outcomes.

While DP steps backward in time, ADP, like RL, steps forward in time. When we step forward in time, we have not computed the value function, so we have to turn to an approximation in order to make decisions.

5.1.2.1 Exogenous Information Process

The system evolves according to several types of exogenous information processes that include random changes to the system parameters i.e., supplies and demand, for example. For complex problems, it is convenient to have a generic variable, w_t to represent all the information that first arrives between $(t-1)$and t. Using S^M to represent a transition function, we represent the evolution of our state variable generically using:

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \tag{5.6}$$

This is called the system model and it indicates the system transition to the next state based on the current state, current decision and the realization of the exogenous information between the current state and the next.

5.1.2.2 ADP Algorithmic Framework

The section provides an overview of the generic ADP framework. Let $\widehat{V}_t(S_t)$ be an approximation of the value function. We assume that we have an initial estimate of $\widehat{V}_t(S_t)$ for each state S_t. Such an approximation introduces error and the challenge is to find approximations that are good enough. ADP proceeds by iteratively estimating the approximation $\widehat{V}_t(S_t)$. The key idea of the ADP framework is to replace the exact value function vector by a statistical approximation in order to overcome the difficulty of dealing with high dimensional state spaces. However, there is still the problem of computing expectation over the random parameters. The second key idea is to use Monte-Carlo samples of the random parameters to simulate a sample path for the system to follow. The approximate sub-problem in step 3 of the framework encapsulates both these ideas. If the value function approximations are close to the true value functions, then the performance of the policy recommended by the approximation should be close to that of the optimal policy. In the next section, we will develop the ADP algorithm for the MPSRP.

The generic framework for ADP is as follows:

Step 1: Initialize the iteration counter for the algorithm by letting n = 1. Choose initial value function approximations for the first iteration, $\hat{V}_t^1(s_t)$.

Step 2: Initialize the time period by letting t = 1. Initialize the state vector $S_1^n$to reflect the initial state of the system.

Step 3: Sample a realization of the exogenous information processes, $w_{t+1}^n$and solve the approximate sub-problem.

$$(x_t^n, s_t^n) = \underset{(x_t, s_{t+1}) \in X_t(s_t^n, w_t^n)}{\text{argmax}} c_t . x_t + \hat{V}_{t+1}^n(W_{t+1}) \qquad (5.7)$$

Step 4: Increase t by 1. If $t \leq T$, then go to step 3.

Step 5: Use the information obtained by solving the approximate subproblems to update the value function approximations. The update function uses the Robbins and Monro

(1951) scheme to stochastically approximate the value function vectors. It can be viewed as a function that maps the value function approximations, the state vectors, the realization of the random parameters at iteration *n* to the value function approximations at iteration *n* +1.

Step 6: Increase *n* by 1 and go to step 2.

5.2 ADP Algorithm for the MPSRP

The ADP algorithm uses the SDP or MDP formulation of the MPSRP presented in Section 4.3 of Chapter 5 as its basis. However, instead of solving the cost-to-go function exactly to optimality, the ADP algorithm approximates the value of the contributions from future stages in order to overcome the curses of dimensionality and make the problem tractable. The subsequent sections outline the details of the approximation method used by the ADP algorithm for the MPSRP problem.

5.2.1 Value Function Approximation

We are interested in finding a policy that maximizes the expected contribution over all the time periods. By the principle of optimality, we can find the optimal policy by solving:

$$V_t(S'_{rjt}, S''_{jt}) = \mathbb{E}\{V_t(S'_{rjt}, S''_{jt}, R_{rt}, J_{jt}) | S'_{rjt}, S''_{jt}\} \tag{5.8}$$

Where,

$$V_t(S'_{rjt}, S''_{jt}, R_{rt}, J_{jt}) = \max_{x_t} \; C_t X_t + V_{t+1}(S'_{rjt+1}, S''_{jt+1}) \tag{5.9}$$

We replace the value function V_{t+1} with a suitable approximation denoted by $\widehat{V}_{t+1}$. Now we solve the following problem for *one* Monte Carlo sample of R_{rt} & J_{jt} (denoted by $\widehat{R}_{rt}$ & $\hat{J}_{jt}$):

$$\tilde{V}_t\left(S'_{rjt}, S''_{jt}, \hat{R}_{rt}, \hat{J}_{jt}\right) = \max_{x_t} \; C_t X_t + \hat{V}_{t+1}\left(S'_{rjt+1}, S''_{jt+1}\right) \tag{5.10}$$

The above problem is referred to as the **approximate subproblem for time period *t***. We let $\tilde{V}_t\left(S'_{rjt}, S''_{jt}, \hat{R}_{rt}, \hat{J}_{jt}\right)$ be the optimal objective value of the approximate subproblem. Starting with a set of value function approximations and an initial state vector, we sequentially solve one subproblem for each time period using one sample of R_{rt} & J_{jt}.

We have to devise a method for solving (5.10) to update and improve the value function approximations $\hat{V}_t$. After the updating procedure, we obtain a new set of value function approximations. Then we solve all the subproblems using the new value function approximations and new sample realizations.

5.2.2 Linear Value Function Approximation

We take our value function approximations to be

$$\hat{V}_t\left(S'_{rjt}\right) = \sum_r \sum_j \hat{V}_{rjt}\left(S'_{rjt}\right) \tag{5.11}$$

$$\hat{V}_t(S''_{jt}) = \sum_j \hat{V}_{jt}(S''_{jt}) \tag{5.12}$$

Where each $\hat{V}_{rjt}$ is a linear function $\hat{V}_{rjt}(S'_{rjt}) = \hat{v}_{rjt}S'_{rjt}$. Similarly, each $\hat{V}_{jt}$ is a linear function $\hat{V}_{jt}(S''_{jt}) = \hat{v}_{jt}S''_{jt}$. Then the approximate subproblem (5.12) can be written as:

$$\tilde{V}_t(S'_{rjt}, S''_{jt}, \hat{R}_{rt}, \hat{J}_{jt}) =$$

$$max \sum_{r\in R}\sum_{j\in J} IWF_{rj}x_{rjt} + \sum_{j\in J} CWF_j\, y_{jt} - \sum_{r\in R} c_r^i\, I_{rt} - \sum_{j\in J} c_\delta^j\, \delta_{jt}$$
$$+ \left(\sum_{r\in R}\sum_{j\in J} \hat{v}_{rjt+1}\, S'_{rjt+1}\right) + \left(\sum_{j\in J} \hat{v}_{jt+1}\, S''_{jt+1}\right)$$

But,

$$S'_{rjt+1} = x_{rjt} \qquad \forall\ j\epsilon J, r\epsilon R$$

$$S''_{jt+1} = y_{jt} \qquad \forall\ j\epsilon J$$

Hence we rewrite (5.14) as:

$$\tilde{V}_t(S'_{rjt}, S''_{jt}, \hat{R}_{rt}, \hat{J}_{jt}) =$$

$$max \sum_{r \in R} \sum_{j \in J} IWF_{rj} x_{rjt} + \sum_{j \in J} CWF_j \, y_{jt} - \sum_{r \in R} c_r^i \, I_{rt} - \sum_{j \in J} c_\delta^j \, \delta_{jt} + \left(\sum_{r \in R} \sum_{j \in J} \hat{v}_{rjt+1} \, x_{rjt} \right)$$
$$+ \left(\sum_{j \in J} \hat{v}_{jt+1} \, y_{jt} \right)$$

Therefore, the approximate subproblem at time period t can be defined as:

$$max \sum_{r \in R} \sum_{j \in J} (IWF_{rj} + \hat{v}_{rjt+1}) x_{rjt} + \sum_{j \in J} (CWF_j + \hat{v}_{jt+1}) y_{jt} - \sum_{r \in R} c_r^i \, I_{rt}$$
$$- \sum_{j \in J} c_\delta^j \, \delta_{jt} \quad (5.13)$$

5.2.3 Updating Value Function Approximations

Let us assume that at iteration n, $\hat{R}_{rt}^n$ is the sequence of supply realizations, $\hat{J}_{jt}^n$ is the sequence of demand realizations. Let $\hat{V}_t^n$ be the sequence of value function approximations. Let $S_{rjt}'^n$ and $S_{jt}''^n$ be the sequence of system states generated by solving approximate subproblems of the following form by using current value function approximations, supply realizations and demand realizations:

$$\tilde{V}_t(S'_{rjt}, S''_{jt}, \hat{R}_{rt}, \hat{J}_{jt}) =$$
$$max \sum_{r \in R} \sum_{j \in J} (IWF_{rj} + \hat{v}_{rjt+1}) x_{rjt} + \sum_{j \in J} (CWF_j + \hat{v}_{jt+1}) y_{jt} - \sum_{r \in R} c_r^i \, I_{rt}$$
$$- \sum_{j \in J} c_\delta^j \, \delta_{jt} \qquad (5.14)$$

At each period our objective is to approximate the value of each feasible state. At each period, the VF approximation of the next state is calculated by the approximate subproblem. In order to get VF approximations of other feasible states, we calculate the reduced costs of each feasible resource – job assignment pair. For linear approximations, the VF approximation of each state is described by a single slope. At each period, we change the state variable and rerun the approximate subproblem for each feasible state. The change in objective function value is the contribution of each feasible state. We use e_{rj} and e_j to denote the modification of the state variables and rerun the approximate subproblem as shown below:

$$\varphi_t^n(e_{rj}) = \tilde{V}_t(S'_{rjt} \sim e_{rj}, \hat{R}_{rt}, \hat{J}_{jt}) - \tilde{V}_t(S'_{rjt}, \hat{R}_{rt}, \hat{J}_{jt}) \quad (5.15)$$

$$\varphi_t^n(e_j) = \tilde{V}_t(S''_{jt} \sim e_j, \hat{R}_{rt}, \hat{J}_{jt}) - \tilde{V}_t(S''_{jt}, \hat{R}_{rt}, \hat{J}_{jt}) \quad (5.16)$$

$\varphi_t^n(e_{rj})$ and $\varphi_t^n(e_j)$ can be likened to the reduced cost of each assignment. It is an estimate of how much the objective function will change when the state variable changes. We assume each linear value function approximation component ($\hat{V}_{rjt}^n$ and $\hat{V}_{jt}^n$) is characterized by slopes $\hat{v}_{rjt}^n$ and $\hat{v}_{jt}^n$ respectively. We update our estimate of the value function approximation using the following equation to obtain the slope of the value function approximation component $\hat{V}_{rjt}^{n+1}$and $\hat{v}_{jt}^{n+1}$. α^n is the step size at iteration n:

$$\hat{v}_{rjt}^{n+1} = (1 - \alpha^n)\hat{v}_{rjt}^n + \alpha^n \varphi_t^n(e_{rj}) \quad (5.17)$$

$$\hat{v}_{jt}^{n+1} = (1 - \alpha^n)\hat{v}_{jt}^n + \alpha^n \varphi_t^n(e_j) \quad (5.18)$$

5.3 ADP Algorithmic Framework for the MPSRP

Step 1: Initialize the iteration counter for the algorithm by letting n = 1. Choose initial value function approximations for the first iteration, $\hat{V}_t^1(S'_{rjt})$ and $\hat{V}_t^1(S''_{jt})$.

Step 2: Initialize the time period by letting t = 1. Initialize the state vector S'^{n}_{rjt} and S''^{n}_{jt} to reflect the initial state of the system.

Step 3: Sample a realization of the exogenous information processes, $\hat{R}_{rt}, \hat{J}_{jt}$ and solve the approximate sub-problem for t.

$$max \sum_{r\in R}\sum_{j\in J}(IWF_{rj}+\hat{v}_{rjt+1})x_{rjt} + \sum_{j\in J}(CWF_j+\hat{v}_{jt+1})y_{jt} - \sum_{r\in R} c_r^i \, I_{rt} - \sum_{j\in J} c_\delta^j \, \delta_{jt}$$

Step 4: Increase t by 1. If $t \leq T$, then go to step 3.

Step 5: Use the information obtained by solving the approximate subproblems to update the value function approximations. It can be viewed as a function that maps the value function approximations, the state vectors, the realization of the random parameters at iteration n to the value function approximations at iteration n +1.

$$\hat{v}_{rjt}^{n+1} = (1-\alpha^n)\hat{v}_{rjt}^n + \alpha^n \, \varphi_t^n(e_{rj}) \quad (5.19)$$

$$\hat{v}_{jt}^{n+1} = (1-\alpha^n)\hat{v}_{jt}^n + \alpha^n \, \varphi_t^n(e_j) \quad (5.20)$$

Step 6: Increase iteration counter n by 1 and go to step 2.

5.4 Alternative ways to update the value function

The commonly used method to update the value function is the Robbins and Monro (1951) stochastic approximation scheme. This scheme is the same as the simple exponential smoothing technique without trend, seasonal components and adaptive mechanisms. Such a model uses only the historical information of the time series (value function approximations in our case) to estimate future values. There are alternate forms of exponential smoothing models that can also be considered in our update function. The following are some forms:

- Holt's Model:
    - This is the simple exponential smoothing model with a linear trend added in. The trend is the average rate of change in the value function approximation from one period to another.
- Winter's Additive Model:
    - If the value function approximations are subject to an additive seasonal factor, for example, increase in attrition during a specific quarter every year, then Winter's additive model accounts for it. We deseasonalize the time series to remove the impact of seasonality.
- Winter's Multiplicative Model:
    - This model is similar to that of the previous model, except that this model accounts for multiplicative seasonal factors that impact the time series (i.e., value function approximations).

5.5 Rollout Algorithms

Other solution methodologies applied to stochastic dynamic problems are heuristic based rollout algorithms (Bertsekas & Castanon, 1999; Bertsekas, Tsitsiklis, & Wu, 1997). Rollout algorithms are based on the policy iteration methods of DP as opposed to value iteration that is used is reinforcement learning and OR based ADP algorithms. These algorithms use heuristic versions of policy iteration to approximate the cost-to-go function which are used to guide decision making in the current state. Rollout policies are implemented within an ADP algorithm that looks ahead one step and solves the subproblem using a heuristic.

From a current state and for a given action, the one-step rollout policy transitions to all possible states that might be observed at the next stage of the problem. From each pre-decision state we execute the heuristic to obtain a policy along with its value. In a one-step rollout algorithm, the estimate of the cost-to-go function when selecting an action in a state is the expected value of the policies obtained in all possible states at the next decision point. For each feasible action a, one-step rollout executes the heuristic $|(s, a)|$ times (where is the s is the number of states). Hence one-step rollout still suffers from the curse of dimensionality and will not be applicable to large problems. Another characteristic of rollout algorithms that differentiates them from ADP algorithms is that they use a heuristic, as opposed to a mathematical model, at each decision point.

5.6 The Information Observation Process & Sequence of Management Action

The ADP solution procedure can be implemented in different ways based on the manner in which resource planning decisions are structured in practice. Specifically, it is related to the sequence of observing realized information about the state of the system and decision making. In the MPSRP, information about available resources and realized jobs is observed in each period of the planning horizon. The management decision involved is that of assigning available internal resources or contingent resources to the realized jobs in each period in order to maximize their contribution. Hence there are two steps in the resource planning process (a) information observation, and (b) management action. There are three different ways in which information observation and management actions can be sequenced in each period.

5.6.1 Observe information first & make decisions

In this method, resource and job availabilities are observed first at the beginning of each period. Based on the observed realizations, management decisions are implemented to match resources to jobs. This approach is referred to as the "wait-and-see" approach where no planning is considered and assignment decisions are made on realizations of resource and job availabilities. This is the ideal situation where the decision maker makes resource assignments with perfect information. However, such a situation is not realistic as the decision maker needs to plan for resources to fill job requirements and decisions have to be made before information is observed.

5.6.2 Make decisions first and observe information

In this case, resource planning decisions are made prior to information observation and the assignments are planned based on the decision maker's point estimates. Sample data are used to calculate point estimates, such as the mean availabilities of resources, and they serve as a best estimate of the random parameter. Such a scenario might arise when the resource planning is completed prior to the start of the planning horizon and actual realizations of information cannot be observed. The accuracy of decisions made in this method depends on the accuracy of the point estimates.

5.6.3 Delayed observation of information

Information about certain features of the problem setting may not be observable at the beginning of each period in the planning horizon. For example, information about the availability of resources at the beginning of each time period can be incomplete as attrition can occur during the course of that period. The decision maker will have accurate information about resource availability only at the end of the period. Job availability, on the other hand, is different – jobs that are already won will be available to be staffed over their duration in the planning horizon. In such a case, at the beginning of each period the decision maker will be able to observe job availability but not resource availability. Such a scenario is labeled as a "resource planning" approach where planning decisions have to be made before observing all the information needed to make decisions. Hence assignment decisions for the current period have to be made based on either the availabilities of resources in the previous period or their point estimates for the current period. In this thesis we focus on developing an ADP algorithm for the case with resource information delay.

5.7 ADP Training & Testing Phases

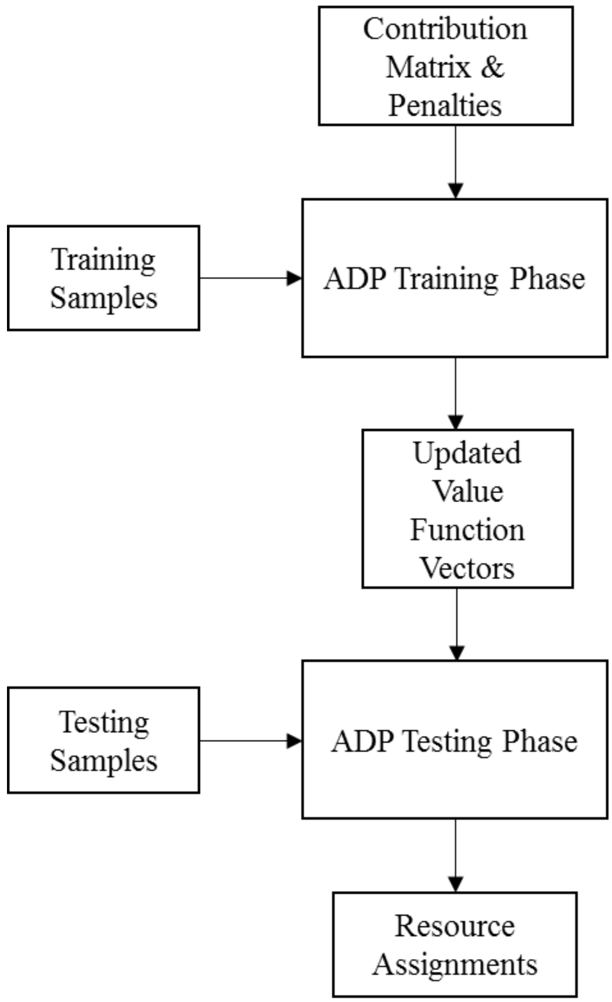

Figure 5.1 – Outline of the ADP Algorithmic Framework

This section presents an outline of the ADP training and testing phases. Phase 1 of ADP is a training phase where the algorithm is trained, over multiple iterations, using the static assignment contributions and Monte Carlo samples of resource and job availabilities. In each iteration of the training phase, a N-period subproblem is built using the Monte Carlo sample for that iteration and the updated value function vectors from the previous iteration (the first iteration uses only the contribution matrix). Let's refer to the solution value of the subproblem as SUB. In order to update the contribution of each resource-job pair, a sensitivity model is run to get the reduced cost of each pair. The updated contribution of each resource-job pair is required to approximate the cost-to-go function of the Bellman equation. Let's refer to the solution value of the sensitivity run as SEN. The sensitivity run is designed as follows: the optimal solution from the subproblem (in each period for each iteration) is our reference solution. In order to obtain the reduced cost of each available resource-job pair, we either turn ON or turn OFF each pair in the subproblem and run the sensitivity model. That is, for optimal resource-job assignments which would be ON in the subproblem, we turn it OFF in the sensitivity run. Similarly, for sub-optimal resource-job assignments which would be OFF in the subproblem, we turn it ON in the sensitivity run. The reduced cost is calculated as seen in equation (5.18):

$$\varphi_t^n(e_{rj}) = \tilde{V}_t(S_{rjt} \sim e_{rj}, \hat{R}_{rt}, \hat{J}_{jt}) - \tilde{V}_t(S_{rjt}, \hat{R}_{rt}, \hat{J}_{jt}) \tag{21}$$

For the ON runs, we calculate:

$$\varphi_t^n(e_{rj}) = SEN - SUB \tag{22}$$

For the OFF runs, we use:

$$\varphi_t^n(e_{rj}) = SUB - SEN \tag{23}$$

This is because the ON runs measure decrease in contribution due to a suboptimal assignment pair. The pair is OFF in the subproblem and is turned ON in the sensitivity run. The SEN objective value will reduce due to the suboptimal pair being forced on and will be lower than the subproblem objective value. In contrast, the OFF runs measure gain in contribution due an optimal assignment pair. The pair is ON in the subproblem and is turned OFF in the sensitivity run. The SEN objective value will decrease and we are able to measure the reduced cost of the optimal pair. This is done for each available resource-job assignment pair in the period for the specific sample path being used.

Resource-Job Pair Contribution	Value Function Update Mechanism
Optimal	Turn OFF, get reduced cost & REWARD pair
Sub-optimal	Turn ON, get reduced cost & PENALIZE pair

Table 5.1 – Value Function Update Mechanism

In this manner, the ADP algorithm is trained via each of the sample paths and is used to update the value functions of each resource-job pair. $\varphi_t^n(e_{rj})$, as seen in (5.18), is used to obtain the reduced cost of each resource-job pair and to approximate their contributions and update their value functions. The updated value function vectors at the end of the training phase is input to the testing phase. New Monte Carlo testing samples are input to the testing phase along with the value function vectors to provide resource planning decision support.

5.8 Summary

Reinforcement learning in artificial intelligence and ADP in operations research provides approximate algorithmic frameworks for these problems. Contrasting with RL, the ADP methodology found in the OR area is heavily based on mathematical programming methods. While RL methods are often labeled as “model-free” to indicate that they do not need the theoretical MDP model, we extend that definition to include mathematical programming models as well. While RL depends on function fitting methods such as regression and neural networks (Gosavi, 2003) to define the approximate sub-problem, the ADP framework develops and solves mathematical models (Powell & Topaloglu, 2006). The advantages of the ADP framework, based on mathematical models, can be seen from its application to large scale problems from practice (Topaloglu & Powell, 2006). Moreover, since it includes the impact of current decisions on future outcomes, ADP clearly provides better decision support when compared to deterministic models, rolling horizon models and open-loop simulation optimization.

6. COMPUTATIONAL EXPERIMENTS

6.1 Experimental Design

We use a three-level full factorial design to evaluate the performance of the ADP and RH algorithms. We vary four factors in our experiments and their explanations are given in Table 6.1.

Table 6.1 - Factors included in the experimental design

Factor	**Factor Explanation**	**Value**
|J|	The Number of Jobs to be staffed	{15, 30, 50}
|R|	The Number of internal workforce resources available	{5, 10, 20}
RP	Job Reassignment Penalty	{10%, 25%, 50%}
IP	Internal Resource Idle Penalty	{50%, 75%, 100%}

The size of the MPSRP problem is influenced by the number of jobs |J|, the number of internal workforce resources |R| and the planning horizon. In our experiment, |J| is chosen from the set {15, 30, 50} and |R| is chosen from {5, 10, 20}. The planning horizon is fixed to be 8 periods. The job reassignment penalty is a percentage of the job value contribution per period. The internal resource idle penalty is a percentage of the IWF resource cost incurred by the company per period. The CWF contribution is set to 25% of the job value contribution per period. The nine size combinations and nine penalty combinations gives rise to a total of 81 experimental combinations.

6.2 Sample Path Generation

Sample paths for resource and job availabilities are generated using Monte Carlo simulation. The market type indicates the value of the jobs that are being bid on. In this dissertation we consider a regular market where 20% of the jobs are low priced, 70% are medium priced and 10% are high priced. The win probabilities of these job categories, in a regular market, are as follows:

Table 6.2 - Job win probabilities

Job Type	Win Probability	Percentage (Regular Market)
Low Priced Job	0.90 - 1	20%
Medium Priced Job	0.70 - 0.90	70%
High Priced Job	0 - 0.70	10%

Based on the stated ranges, the win probability for each job is generated using a uniform distribution. Additionally, each job has a time window randomly generated from a uniform distribution within which it is expected to be won by the company. Within its time window, a job has its specified win probability and it reduces to zero outside of it. The job durations are fixed to 6 periods. The resource attrition probabilities are as follows:

Table 6.3 - Internal resource attrition probabilities

Resource Type	Attrition Probability	Percentage
Low Attrition Resources	0 - 0.10	20%
Medium Attrition Resources	0.10 - 0.25	70%
High Attrition Resources	0.25 - 0.35	10%

We generate 100 training sample paths and 100 testing sample paths using Monte Carlo simulation for both resource and job availabilities. ADP is first trained using the training sample paths and the updated value function vectors from the training algorithm are tested using the testing sample paths. We use the step size α^n = 20/(40 + n) at sample path n. RH is implemented using the testing sample paths and the point estimates.

6.3 Benchmark Solution Approach: Rolling Horizon

The rolling horizon procedure uses point estimates of future supply and demand realizations. An n-period rolling horizon solves an n-period deterministic IP for every time period. For the first time period we use the actual resource and job realizations of the current sample path at time t and the next n-1 time periods use the expected values of the realizations. Once this IP is solved, we implement decisions of the first time period and proceed to solve the problem for time period t+1 with the boundary conditions changed appropriately.

6.3.1 Generating Point Estimates for Rolling Horizon

Rolling horizon makes use of point estimates for fixing future availabilities of jobs and resources deterministically. We use a threshold of 0.75 for job win probabilities and 0.20 for resource attrition probabilities for the deterministic rolling horizon procedure. For example, if a job's win probabilities is greater than 0.75 the decision maker will assume that job to be won and will include it in his staffing plans. If the job's win probability is less than the decision maker's threshold, the job will be assumed to be lost. IWF resource availability is also determined in a similar manner by the decision maker.

6.3.2 Delayed Observation of Resource Availability

As mentioned earlier in section 5.6.3, this thesis deals with the case where job availabilities are observed at the start of a period but resource availabilities are only completely observed at the end of a period. In this manner we make provision for resource attrition to occur during the course of any planning period.

6.3.2.1 ADP Implementation

We blindfold the ADP testing phase to resource attrition and purely depend upon the updated VF vectors from the training phase to guide the ADP testing phase. After the actual resource availabilities are realized for the period, the assignments are validated. If a job was assigned to an unavailable resource, the job is sent to CWF on an urgent basis. After the post-decision updates are completed for the period, the assignments are fixed and the ADP procedure moves on to the next period.

6.3.2.2 Rolling Horizon Implementation

At the start of each period the job availabilities for that period are observed. Based on the observed information for the current period, the point estimates can be updated. If a job starts this period, its point estimates is updated to be available for the job's duration if it assumes the job to be unavailable. If a job did not start in the current period but the point estimate assumes that it starts, then the point estimate is modified to be unavailable. In this way, the job point estimates are updated at the start of each period. The resource point estimates cannot be updated at the start of a period as accurate resource information is not observable yet. The availability of a resource over a period is only precisely observable at the end of the period. Hence the RH procedure uses the resource point estimates without updating them. The problem is solved, for each period, using the updated job point estimates and the static resource point estimates.

After the current period's problem is solved and the assignments are made, the resource availabilities can be observed at the end of the period. Now, the assignments that were made using the resource point estimates can be validated. There are 3 possible conditions based on the actual resource realizations:

1. The resource point estimate assumes that a resource is unavailable while in reality the resource was available to be staffed. In this case, the resource is considered to be idle and an idle penalty is imposed on the objective value.
2. The resource point estimate assumes that a resource is available and assigns it to a job. However, the resource is unavailable in reality. In this case, the assignment is considered to be invalid. The job is sent to the CWF on an urgent basis.
3. The point estimates assumes that a resource is available but leaves the resource unassigned, thus incurring an idle penalty. If the resource is not available in reality, then the idle penalty is removed.

In this manner, after the assignments are made in each period, they are updated based on actual realizations of resource information. After the update is completed, the current period's assignments are fixed and the RH proceeds to solve the next period's problem following the same procedure as the previous period.

6.4 Computational Results

The algorithms are implemented in ILOG CPLEX 12.5.1. The experiments were run on two different machines. The ADP training phase was run on a machine with an Intel core i-7 processor at 3.40 GHz with 32 gigabytes of RAM. The ADP testing phase and the RH procedure was run on a machine with an Intel core i-5 processor at 2.50 GHz and 16 gigabytes of RAM.

6.4.1 Summary of Key Observations

No. of Resources	No. of Jobs	RH Mean Obj. Value		ADP Mean Obj. Value		Mean ADP - RH Gap	Mean Relative Percentage Gap
		Mean	Standard Deviation	Mean	Standard Deviation		
5	15	$2,555,121	$270,638	$3,366,253	$592,634	$811,132	31.75%
5	30	$4,810,941	$270,159	$5,837,759	$723,361	$1,026,818	21.36%
5	50	$9,151,764	$278,698	$10,236,626	$787,782	$1,084,861	11.82%
10	15	$2,535,074	$539,736	$3,117,662	$679,454	$582,588	23.38%
10	30	$5,041,450	$502,777	$6,360,408	$978,498	$1,318,958	26.23%
10	50	$9,787,938	$545,235	$11,484,262	$1,315,615	$1,696,324	17.21%
20	15	$769,390	$1,118,755	$665,337	$1,140,602	-$104,052	-20.20%
20	30	$5,882,393	$919,539	$6,253,479	$1,184,688	$371,086	5.97%
20	50	$11,570,076	$1,015,648	$12,625,090	$1,934,410	$1,055,014	8.74%

Table 6.4 - Summary Results of Computational Experiments by Problem Size

OBSERVATION 1: The ADP algorithm outperforms the RH procedure in 8 of the 9 size combinations. RH performs better than ADP in the case where there are a greater number of IWF resources than jobs that need to be staffed. Table 6.4 exhibits the summary results by problem size. RH performs well when demand is low and supply is high. Upon investigation, we found that, in this case, RH relies less on CWF, more on low risk IWF resources and incurs less job reassignment penalty.

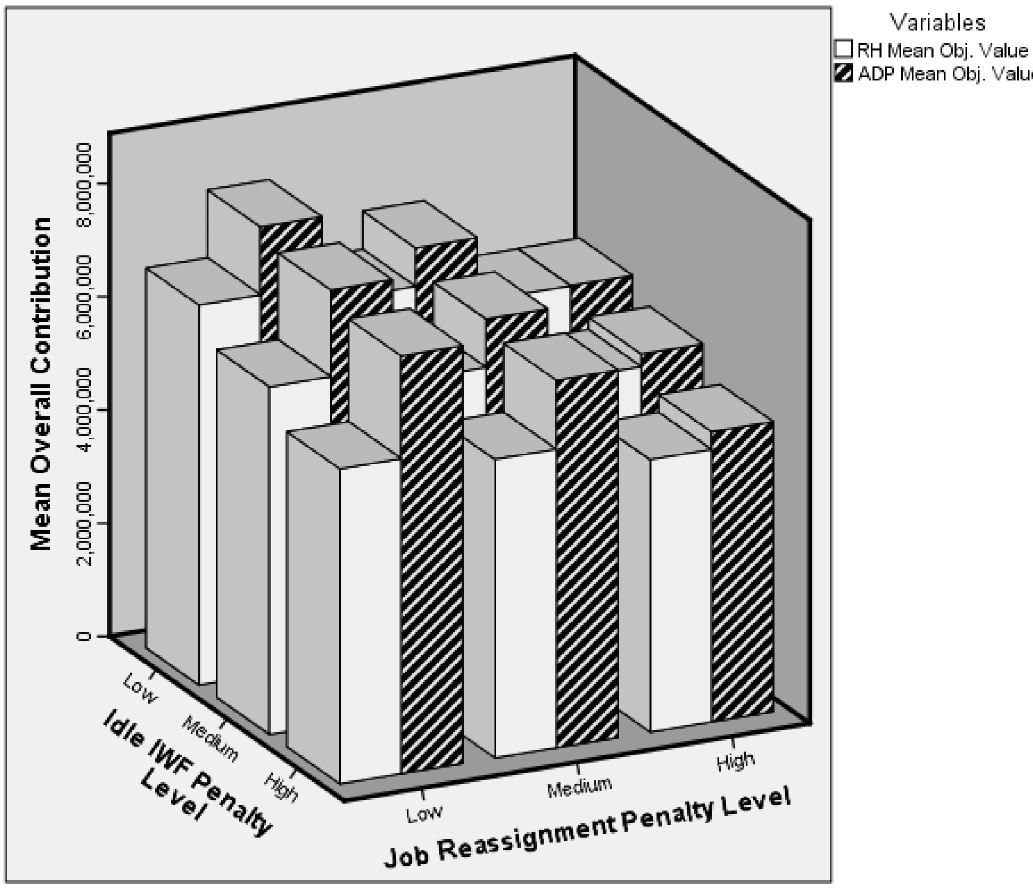

Figure 6.1 - Mean Objective Values by Reassignment Penalty & Idle Penalty

OBSERVATION 2: There is an inherent trade-off between the job reassignment penalty and the idle IWF resource penalty. RH incurs higher idle IWF resource penalty and ADP incurs higher job reassignment penalty.

Figure 6.1 shows the difference in mean objective values for ADP and RH over the various penalty combinations. The gap between ADP and RH can be referred to as the ADP-RH gap and it indicates the extent to which ADP outperforms RH. The ADP-RH gap decreases when job reassignment penalty increases and increases when IWF idle penalty increases. RH is marginally better than ADP when reassignment penalty is high and idle penalty is low. Tables 6.5 through 6.7 provide detailed results broken down by the reassignment penalty level and they clearly show the trade-off between the idle penalty level and the job reassignment penalty level. The rolling horizon procedure incurs higher idle IWF resource penalty while ADP incurs higher job reassignment penalty. The experiments have been setup in a way that job reassignments are unavoidable. That is, since the number of jobs are greater than the number of IWF resources and IWF resource attrition is inevitable, job reassignments and the use of CWF is required. The two procedures differ in how they handle this situation and it is discussed in the next observation.

Job Reassignment Penalty	Idle IWF Resource Penalty	No. of IWF Resources	No. of Jobs	RH Mean Obj. Value	ADP Mean Obj. Value	Relative Gap
LOW	LOW	5	15	$2,964,670	$3,987,880	34.51%
LOW	MED	5	15	$2,687,340	$4,029,550	49.95%
LOW	HIGH	5	15	$2,410,010	$3,955,400	64.12%
LOW	LOW	10	15	$3,378,920	$3,969,470	17.48%
LOW	MED	10	15	$2,864,040	$3,718,600	29.84%
LOW	HIGH	10	15	$2,345,620	$3,401,290	45.01%
LOW	LOW	20	15	$2,325,760	$2,214,210	-4.80%
LOW	MED	20	15	$1,071,910	$947,796	-11.58%
LOW	HIGH	20	15	-$180,211	-$303,116	-68.20%
LOW	LOW	5	30	$5,208,680	$6,636,300	27.41%
LOW	MED	5	30	$4,920,120	$6,647,960	35.12%
LOW	HIGH	5	30	$4,631,560	$6,620,350	42.94%
LOW	LOW	10	30	$5,818,390	$7,419,540	27.52%
LOW	MED	10	30	$5,336,410	$7,468,950	39.96%
LOW	HIGH	10	30	$4,850,000	$7,426,740	53.13%
LOW	LOW	20	30	$7,350,950	$8,014,120	9.02%
LOW	MED	20	30	$6,534,640	$7,335,480	12.26%
LOW	HIGH	20	30	$5,669,450	$6,752,730	19.11%
LOW	LOW	5	50	$9,577,930	$11,096,200	15.85%
LOW	MED	5	50	$9,297,670	$11,178,200	20.23%
LOW	HIGH	5	50	$9,017,400	$11,207,100	24.28%
LOW	LOW	10	50	$10,631,600	$12,852,200	20.89%
LOW	MED	10	50	$10,147,900	$12,965,300	27.76%
LOW	HIGH	10	50	$9,674,120	$12,822,400	32.54%
LOW	LOW	20	50	$13,158,700	$15,023,100	14.17%
LOW	MED	20	50	$12,405,500	$14,753,300	18.93%
LOW	HIGH	20	50	$11,667,700	$14,522,800	24.47%

Table 6.5 - Mean Objective Value & Gap for Low Reassignment Penalty Level

Job Reassignment Penalty	Idle IWF Resource Penalty	No. of IWF Resources	No. of Jobs	RH Mean Obj. Value	ADP Mean Obj. Value	Relative Gap
MED	LOW	5	15	$2,853,430	$3,601,040	26.20%
MED	MED	5	15	$2,576,090	$3,433,650	33.29%
MED	HIGH	5	15	$2,298,760	$3,329,440	44.84%
MED	LOW	10	15	$2,955,840	$3,598,830	21.75%
MED	MED	10	15	$2,603,520	$3,206,010	23.14%
MED	HIGH	10	15	$2,094,680	$3,051,770	45.69%
MED	LOW	20	15	$2,069,350	$1,959,420	-5.31%
MED	MED	20	15	$827,933	$791,120	-4.45%
MED	HIGH	20	15	-$429,557	-$624,294	-45.33%
MED	LOW	5	30	$5,116,170	$5,878,670	14.90%
MED	MED	5	30	$4,827,610	$5,837,950	20.93%
MED	HIGH	5	30	$4,539,050	$5,959,340	31.29%
MED	LOW	10	30	$5,575,670	$6,608,550	18.52%
MED	MED	10	30	$5,082,630	$6,380,670	25.54%
MED	HIGH	10	30	$4,598,410	$6,354,240	38.18%
MED	LOW	20	30	$6,799,780	$6,944,950	2.13%
MED	MED	20	30	$5,999,080	$6,394,880	6.60%
MED	HIGH	20	30	$5,158,730	$5,905,710	14.48%
MED	LOW	5	50	$9,454,210	$10,299,500	8.94%
MED	MED	5	50	$9,173,950	$10,010,600	9.12%
MED	HIGH	5	50	$8,893,690	$10,247,800	15.23%
MED	LOW	10	50	$10,344,300	$11,595,000	12.09%
MED	MED	10	50	$9,866,600	$11,691,500	18.50%
MED	HIGH	10	50	$9,384,290	$11,799,300	25.73%
MED	LOW	20	50	$12,458,500	$13,259,700	6.43%
MED	MED	20	50	$11,664,200	$12,553,700	7.63%
MED	HIGH	20	50	$10,905,500	$12,399,800	13.70%

Table 6.6 - Mean Objective Value & Gap for Medium Reassignment Penalty Level

Job Reassignment Penalty	Idle IWF Resource Penalty	No. of IWF Resources	No. of Jobs	RH Mean Obj. Value	ADP Mean Obj. Value	Relative Gap
HIGH	LOW	5	15	$2,679,640	$2,815,690	5.08%
HIGH	MED	5	15	$2,401,740	$2,641,950	10.00%
HIGH	HIGH	5	15	$2,124,410	$2,501,680	17.76%
HIGH	LOW	10	15	$2,785,440	$2,833,170	1.71%
HIGH	MED	10	15	$2,220,550	$2,520,380	13.50%
HIGH	HIGH	10	15	$1,567,060	$1,759,440	12.28%
HIGH	LOW	20	15	$1,651,940	$1,638,820	-0.79%
HIGH	MED	20	15	$419,666	$296,878	-29.26%
HIGH	HIGH	20	15	-$832,283	-$932,798	-12.08%
HIGH	LOW	5	30	$4,973,860	$5,114,270	2.82%
HIGH	MED	5	30	$4,684,990	$5,101,600	8.89%
HIGH	HIGH	5	30	$4,396,430	$4,743,390	7.89%
HIGH	LOW	10	30	$5,191,730	$5,265,950	1.43%
HIGH	MED	10	30	$4,697,230	$5,277,480	12.35%
HIGH	HIGH	10	30	$4,222,580	$5,041,550	19.40%
HIGH	LOW	20	30	$5,866,340	$5,797,270	-1.18%
HIGH	MED	20	30	$5,189,300	$4,884,110	-5.88%
HIGH	HIGH	20	30	$4,373,270	$4,252,060	-2.77%
HIGH	LOW	5	50	$9,263,940	$9,477,500	2.31%
HIGH	MED	5	50	$8,983,670	$9,439,310	5.07%
HIGH	HIGH	5	50	$8,703,420	$9,173,420	5.40%
HIGH	LOW	10	50	$9,829,220	$10,107,100	2.83%
HIGH	MED	10	50	$9,346,950	$9,844,530	5.32%
HIGH	HIGH	10	50	$8,866,460	$9,681,030	9.19%
HIGH	LOW	20	50	$11,343,700	$10,621,500	-6.37%
HIGH	MED	20	50	$10,660,600	$10,607,200	-0.50%
HIGH	HIGH	20	50	$9,866,280	$9,884,710	0.19%

Table 6.7 - Mean Objective Value & Gap for High Reassignment Penalty Level

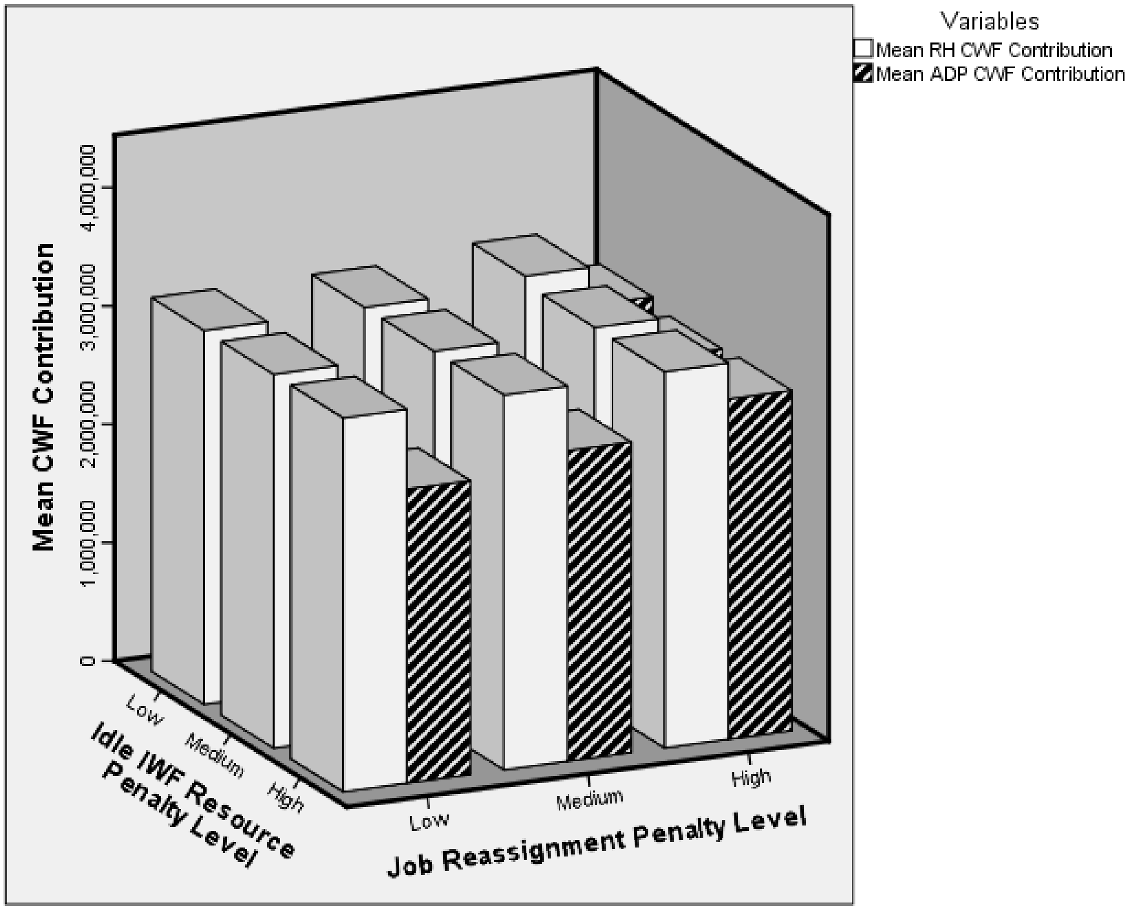

Figure 6.2 - CWF Contribution by Reassignment & Idle Penalty Levels

OBSERVATION 3: ADP utilizes more of the internal workforce to staff the jobs, while RH utilizes more of the external contingent workforce. ADP has higher IWF utilization.

RH discards the high-risk IWF resources and depends more on CWF resources to staff jobs. RH gets a higher level of contribution from outsourcing the jobs to the CWF, especially when the reassignment penalty levels are low as seen in figure 6.2. Indeed, this is evident in the way RH makes use of point estimates. RH takes a safer route through its solution process by discarding high risk IWF resources i.e., resources with higher levels of attrition probability. ADP, on the other hand, includes uncertainty into its solution process and uses more IWF resources than RH which is evident from figure 6.3. ADP does not discard high-risk IWF resources but rather intelligently balances the two penalties. Indeed, it is clear from the results that when job reassignment penalty is low, ADP uses more of IWF resources (which increases the likelihood of job reassignments due to IWF attrition) but reduces dependence on the IWF resources when the job reassignment penalty increases.

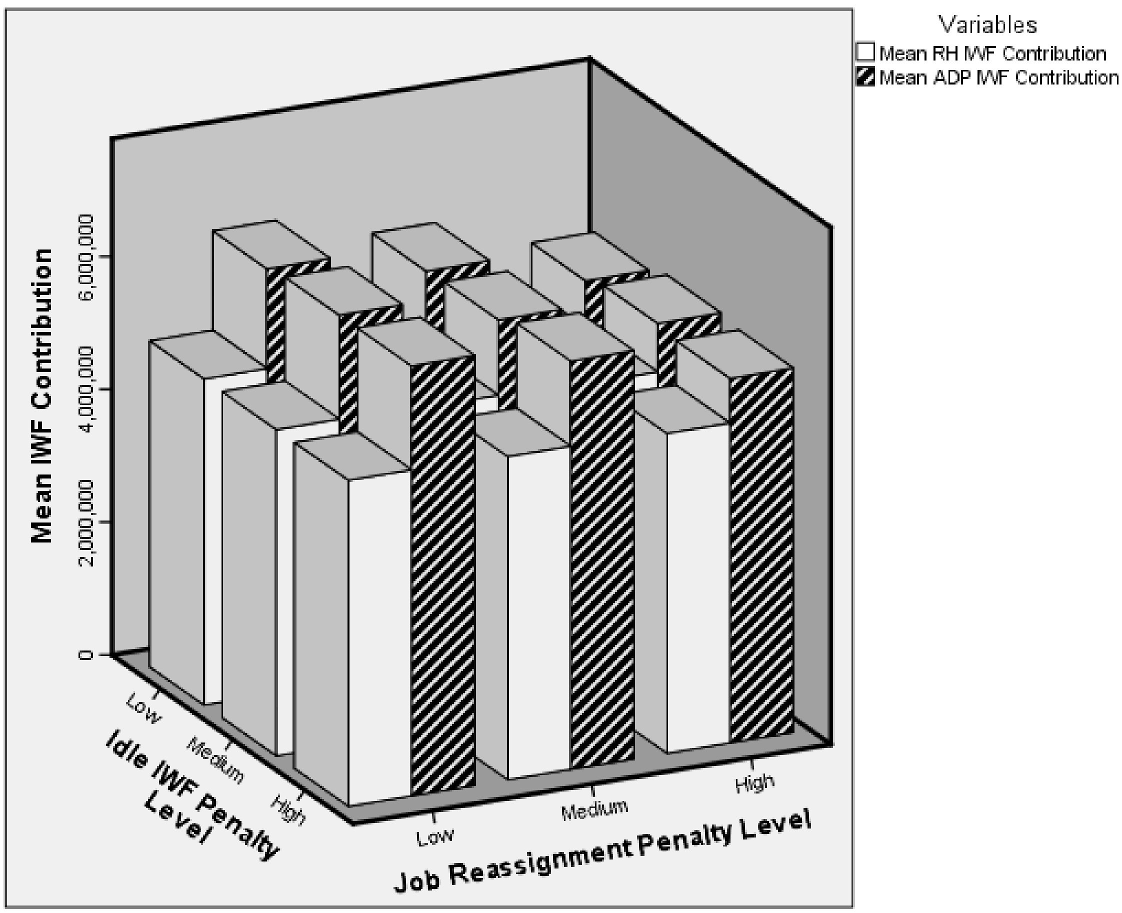

Figure 6.3 - IWF Contribution by Reassignment & Idle Penalty Levels

OBSERVATION 4: RH incurs higher idle IWF resource penalty than ADP. It discards high risk IWF resources.

One consequence of RH sending more jobs to the CWF is that it would have to keep IWF resources idle. This can be seen from figure 6.4 which shows the idle penalty incurred for RH and ADP. From the figure, it is not only clear that ADP incurs less idle penalty than RH, but ADP is intelligent in how it balances the job reassignment penalty and the idle IWF resource penalty. ADP's idle penalty is high when reassignment penalty level is high. This indicates that ADP keeps more IWF resources idle for high reassignment penalty levels i.e., this implies that the jobs that have been sent to the CWF by ADP as a result of IWF attrition are not being brought back to the IWF to avoid the high reassignment penalty. However as reassignment penalty levels reduce, ADP incurs lesser idle IWF penalty indicating that it is reassigning jobs back to the IWF. This shows ADP's balancing act of managing the job reassignment penalty and the idle resource penalty. RH's idle penalty remains fixed regardless of the reassignment penalty which is evidence of the myopic nature of the procedure. This indicates a lack of sensitivity by the RH procedure to the IWF resource attrition. Discarding high risk IWF resources results in suboptimal assignments and higher levels of idle penalty for RH.

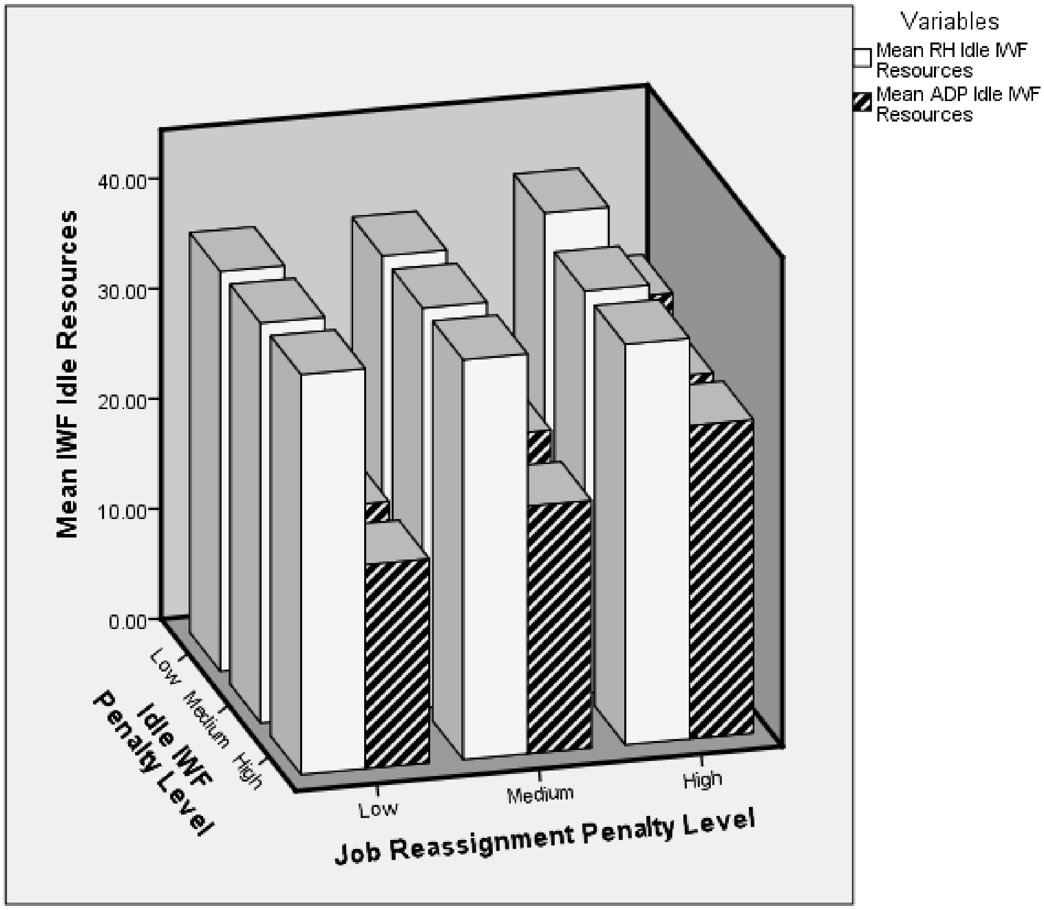

Figure 6.4 - Idle Penalty by Reassignment & Idle Penalty Levels

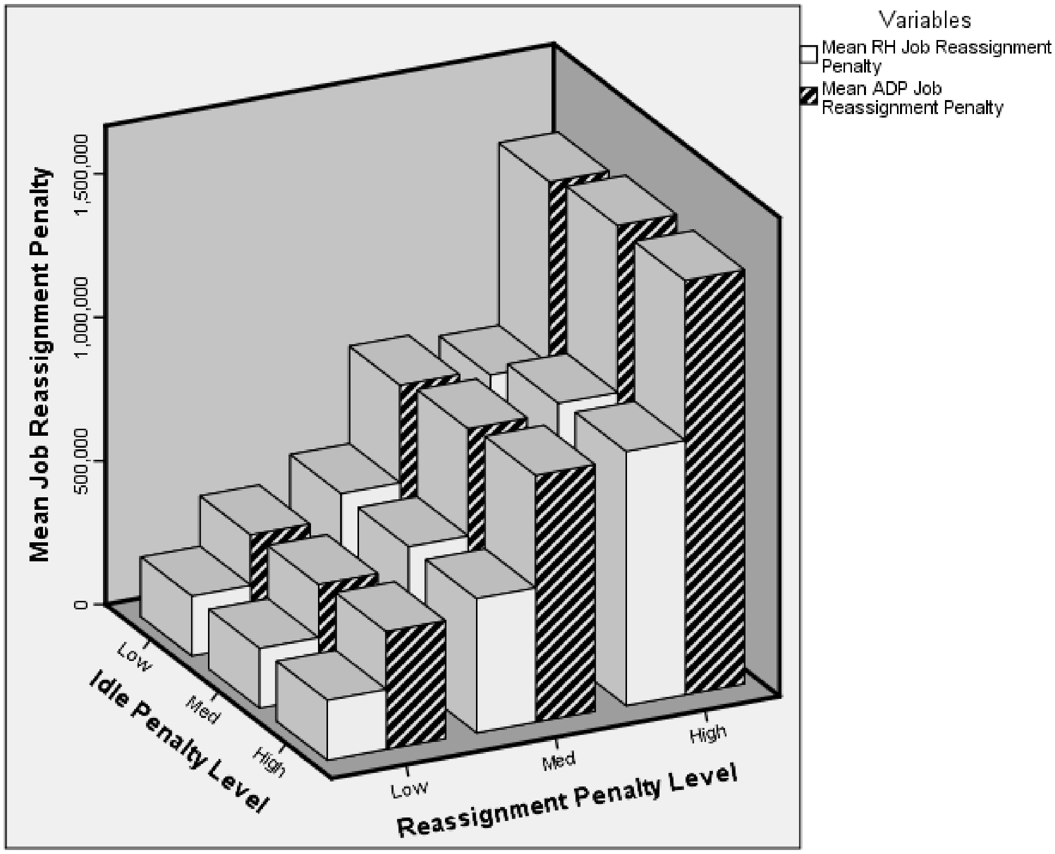

Figure 6.5 - Reassignment Penalty by Reassignment & Idle Penalties

OBSERVATION 5: ADP's propensity to use more IWF resources for staffing the jobs results in a higher number of reassigned jobs.

This is the result of ADP using the updated value function vectors instead of point estimates. The point estimates used by RH discards the high risk resources which can result in lower reassignments but higher idle resources. ADP incurs higher levels of job reassignments but lower levels of idle resources. Figures 6.6 and 6.7 show the resource utilization for RH and ADP respectively. We can observe that RH utilizes similar percentages of IWF and CWF for various penalty levels. That is, its resource mix for staffing is the same regardless of the penalty faced. ADP, on the other hand, balances the use of IWF and CWF resources against the penalty. ADP provides a better resource mix that takes into consideration the IWF attrition levels and the various penalty levels.

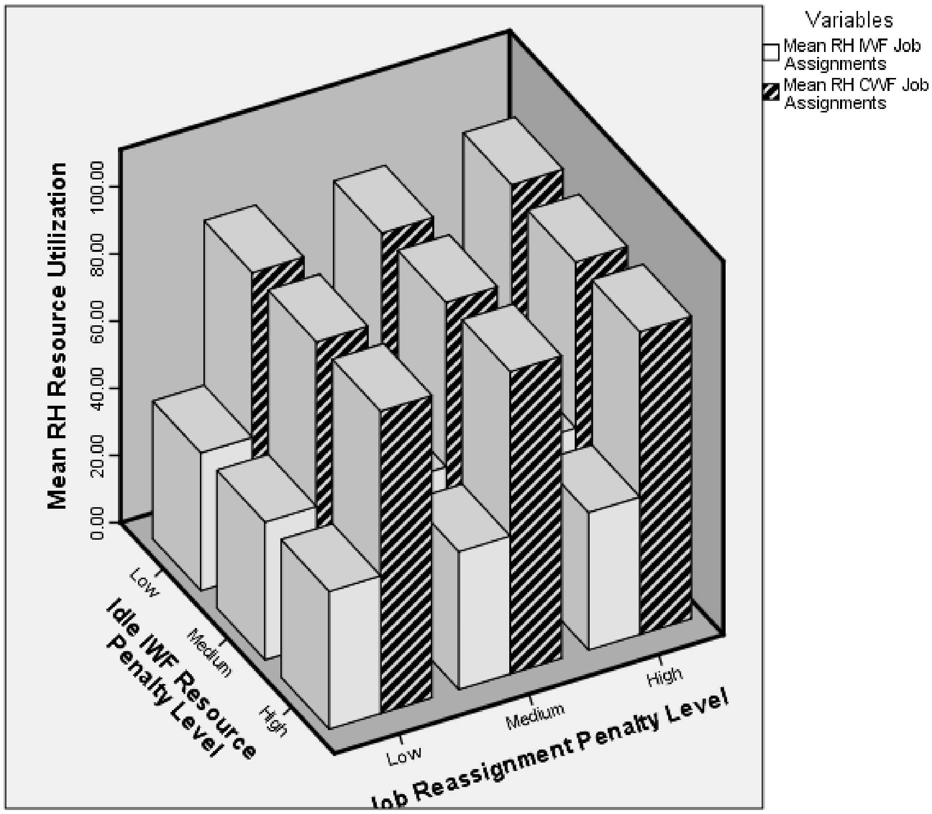

Figure 6.6 - RH Resource Utilization by Reassignment & Idle Penalties

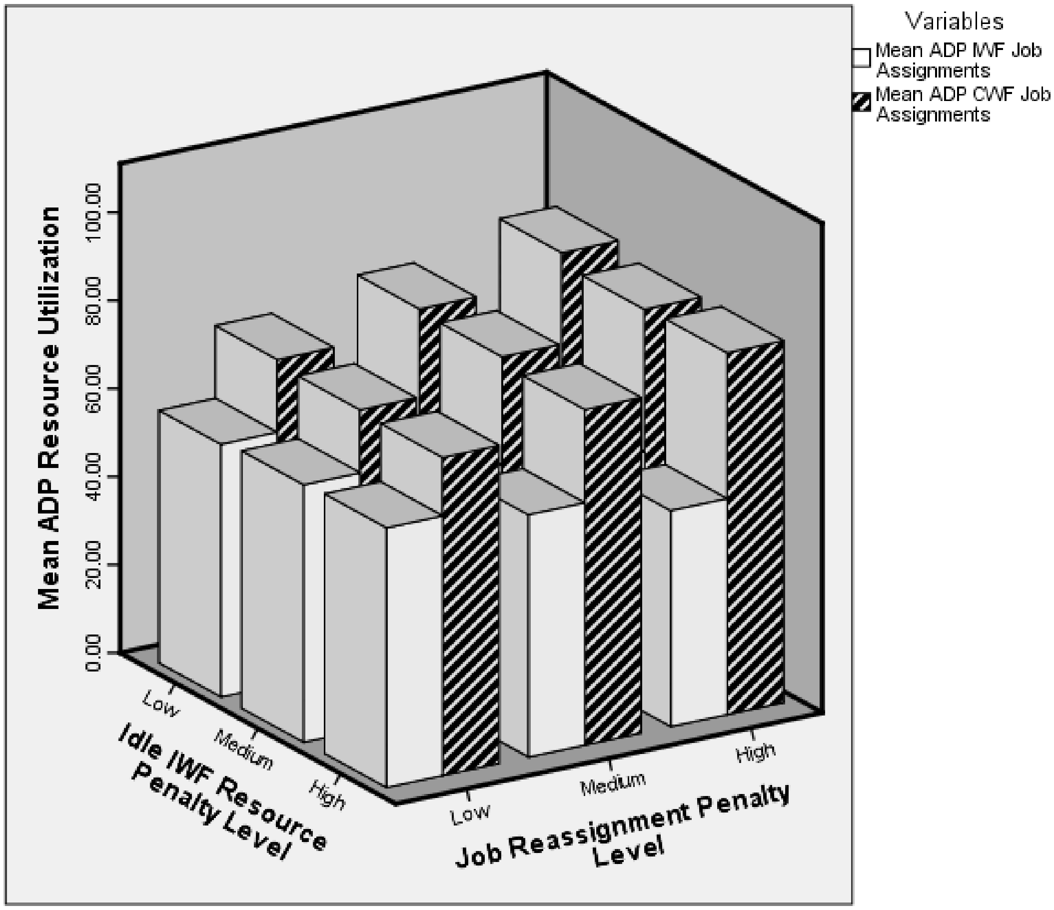

Figure 6.7 - ADP Resource Utilization by Reassignment & Idle Penalties

OBSERVATION 6: As demand increases, ADP's performance benefit over RH improves contingent on penalties

A two-way between-groups analysis of variance was conducted to explore the impact of the number of jobs and reassignment penalty level on the ADP-RH gap. The interaction between the number of jobs and reassignment penalty level was significant ($p < 0.001$). Both the number of jobs and the reassignment penalty level have significant main effects ($p < 0.001$). The interaction plot is shown in Figure 6.8. From the plot, it is clear that the gap reduces when reassignment penalty levels increase. We can also see that the gap is greater when the number of jobs under consideration for staffing increases.

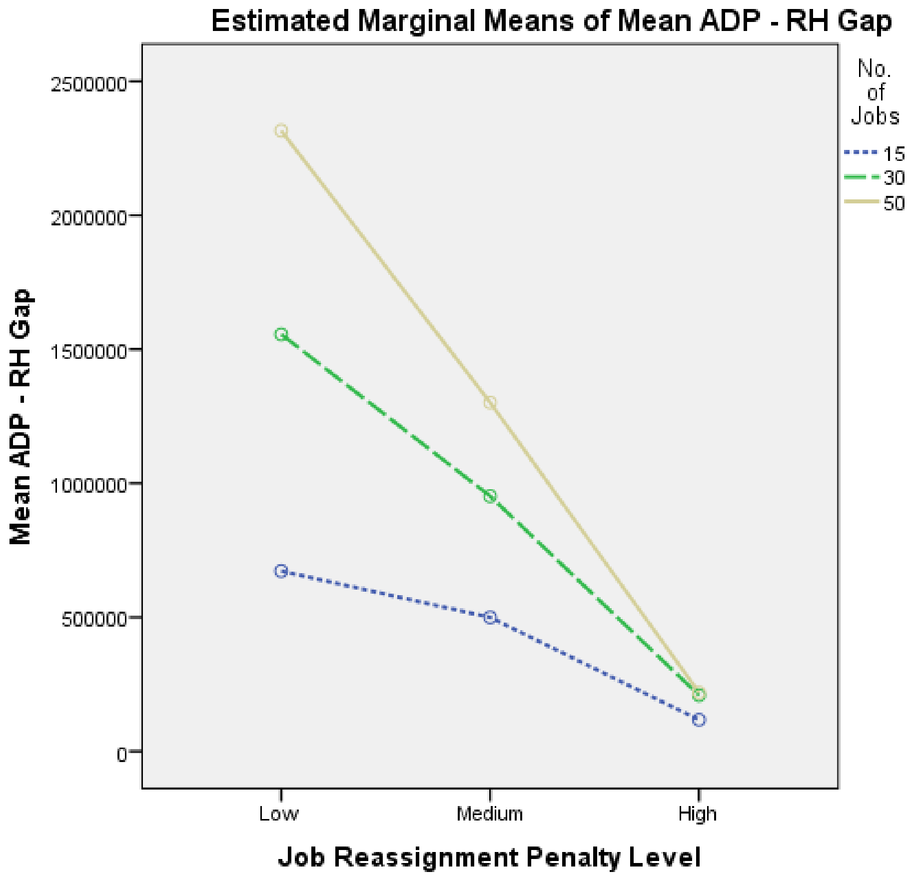

Figure 6.8 - Two Way ANOVA: No. of Jobs & Reassign Penalty on ADP-RH Gap

A second two-way between-groups analysis of variance was carried out to look at the impact of the number of jobs and idle penalty level on the ADP-RH gap. The interaction between the number of jobs and idle penalty level was not significant (p = 0.902). There was a statistically significant main effect for the number of jobs (p = 0.001), but not for the idle penalty level (p = 0.079). From the plot, it is clear that the gap increases as the idle penalty levels increase. We can also see that the gap is greater when the number of jobs to be staffed increases. That is, the contribution from the ADP is greater than RH when the two procedures have a greater number of jobs to contend with.

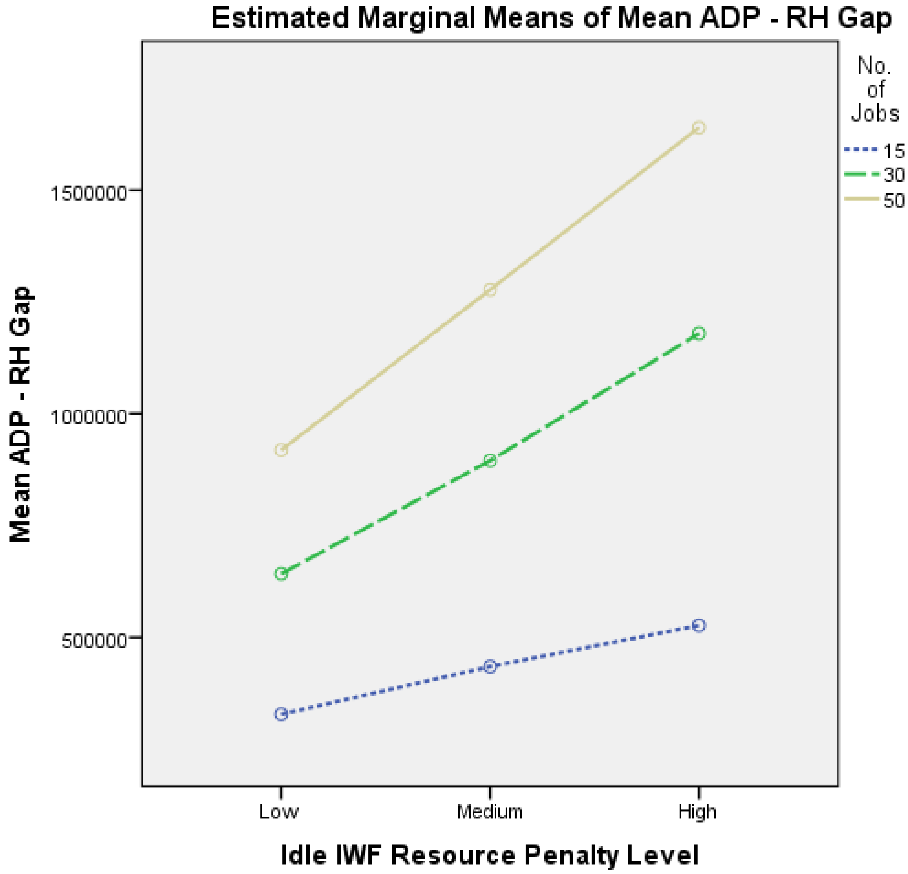

Figure 6.9 - Two Way ANOVA: No. of Jobs & Idle Penalty on the ADP-RH Gap

OBSERVATION 7: ADP's performance stochastically dominates the RH procedure, contingent on the reassignment and idle penalty levels

Figures 6.10 through 6.12 show the ADH-RH gap over the 100 sample paths for the experimental combinations under consideration. These graphs exhibit the performance benefit of ADP over RH over each of the sample paths instead of the average performance over all sample paths. It is clear from the graphs that ADP completely dominates RH when the reassignment penalty is low. The performance degrades when the penalty level increases, however for high levels of idle penalty ADP performance is superior even at high reassignment penalty. For instance, in Figure 6.12 the gap reaches zero at about the 65th percentile for high idle penalty compared to the 37th percentile for low idle penalty. This shows ADP's ability to counteract the job reassignment penalty with the better IWF resource utilization.

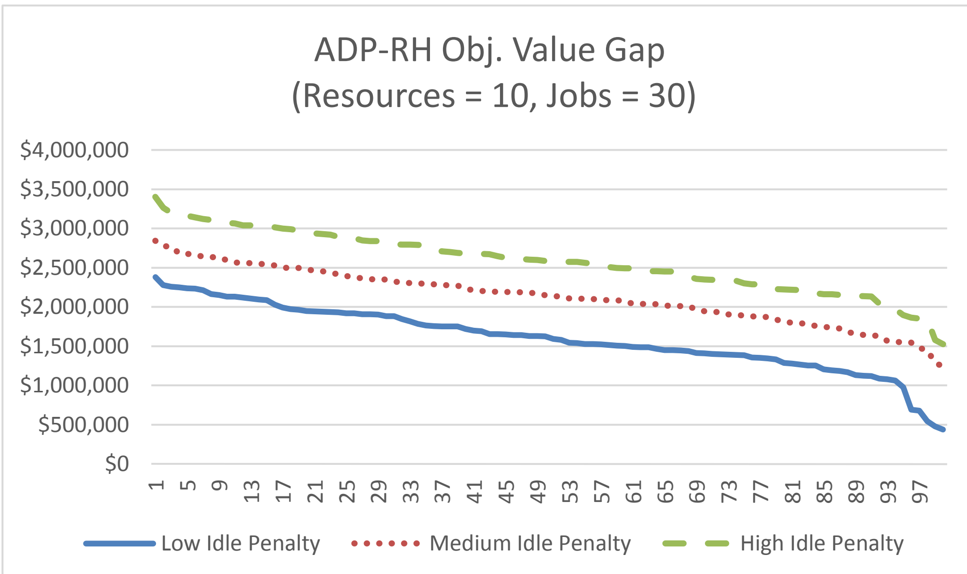

Figure 6.10 - ADP-RH Obj. Value Gap for Low Reassignment Penalty

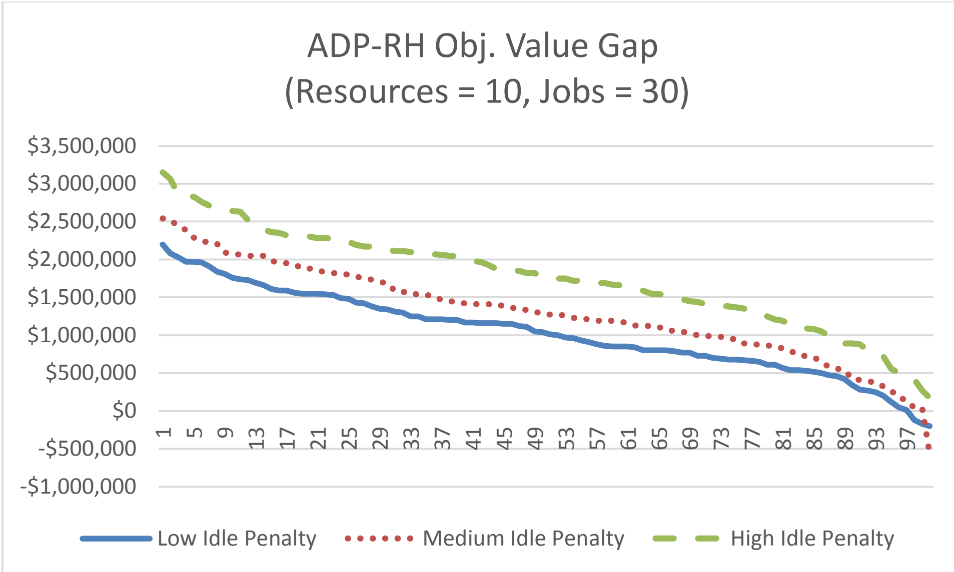

Figure 6.11 - ADP-RH Obj. Value Gap for Medium Reassignment Penalty

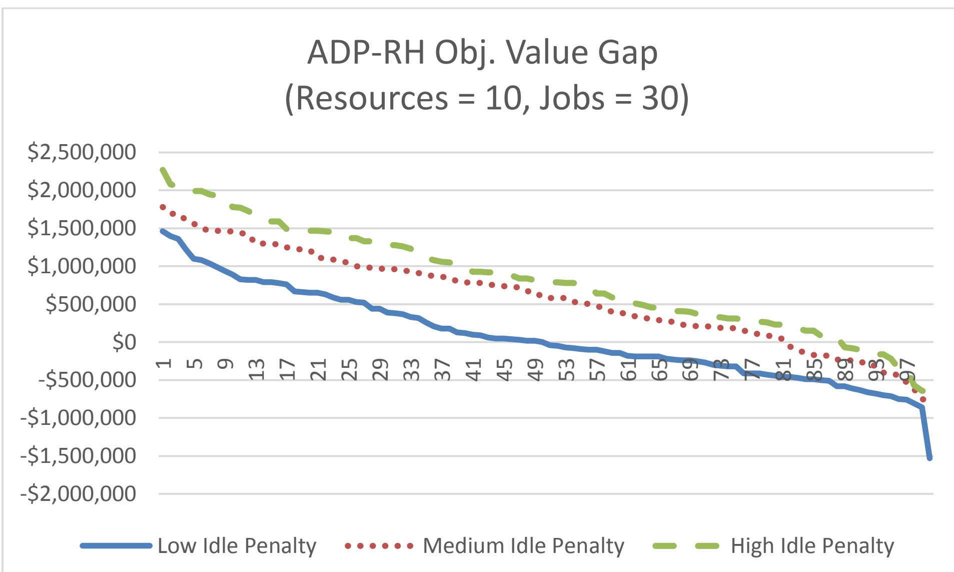

Figure 6.12 - ADP-RH Obj. Value Gap for High Reassignment Penalty

No. of Resources	No. of Jobs	ADP Training Phase (Minutes)	ADP Testing Phase (Minutes)	Rolling Horizon (Minutes)
5	15	22.39	1.03	1.58
10	15	97.56	1.90	3.40
20	15	358.15	3.74	7.10
5	30	88.45	2.04	3.34
10	30	357.90	3.83	7.49
20	30	1609.31	8.45	17.45
5	50	264.57	3.65	6.90
10	50	1013.05	7.58	14.53
20	50	2276.64	16.56	36.89

Table 6.8 - Run times for the ADP phases and the RH procedure

Observation 8: Learning the approximation of the cost-to-go function during the ADP Training Phase is computational intensive.

Idle IWF Resource Penalty Level	Job Reassignment Penalty Level	Job Threshold = 0.75		Job Threshold = 0.50	
		RH Mean Obj. Value	Mean Percentage ADP - RH Gap	RH Mean Obj. Value	Mean Percentage ADP - RH Gap
LOW	LOW	$6,712,844	18.01%	$6,721,476	17.93%
LOW	MED	$6,403,028	11.74%	$6,418,121	11.03%
LOW	HIGH	$5,953,979	0.87%	$5,954,504	0.80%
MED	LOW	$6,140,614	24.72%	$6,146,158	24.80%
MED	MED	$5,846,846	15.59%	$5,841,731	15.94%
MED	HIGH	$5,400,522	2.17%	$5,409,660	1.92%
HIGH	LOW	$5,565,072	26.38%	$5,574,379	27.10%
HIGH	MED	$5,271,506	20.42%	$5,268,729	20.86%
HIGH	HIGH	$4,809,736	6.36%	$4,832,766	5.45%

Table 6.9 - RH Mean Objective Value for different job thresholds

Observation 9: The performance benefit of ADP over RH holds when the job availability point estimate threshold for RH is varied.

The job availability threshold is used by the decision maker to fix future job availabilities for the deterministic RH procedure. In order to test the sensitivity of the RH solution to varying point estimate thresholds, a different RH run was implemented with a job availability threshold of 0.50. The results are summarized and compared with the original threshold of 0.75 in table 6.8. The performance of RH does not vary much with the lower threshold. This is because RH discards the high risk IWF resources and it is unable to improve its IWF utilization as seen from the following table.

No. of Jobs	No. of IWF Resources	Mean RH IWF Job Assignments (Job Threshold = 0.75)	Mean RH IWF Job Assignments (Job Threshold = 0.50)
15	5	14.36	14.36
15	10	32.46	32.59
15	20	54.35	54.26
30	5	14.50	14.50
30	10	35.09	35.02
30	20	79.20	79.40
50	5	14.50	14.50
50	10	35.42	35.39
50	20	85.94	85.73

Table 6.10 - RH IWF resource utilization for varying job availability thresholds

6.5 Comments

We have tested the ADP and RH algorithms on 81 MPSRP computational instances based on a full factorial experimental design. The results clearly show the superiority of ADP over RH in resource planning under uncertainty. ADP excels in solution quality including the objective value and in terms of IWF utilization. The data analysis reveals the trade-off that exists between job reassignment penalty and IWF idle resource penalty. It is these two factors that have the highest impact on algorithmic performance. This is to be expected as both job uncertainty and resource attrition impact these factors and the two algorithms differ in the way they balance the uncertainties and penalties.

ADP takes both job uncertainty and resource attrition into account in its training phase and provides updated value function vectors that reflect the inherent uncertainties. RH, on the other hand, discards high risk resources and does not consider them to be available. This is evident in the way RH sends most of the jobs to the CWF. However, by doing so it incurs a higher level of IWF idle penalty. Assigning jobs to the CWF is a safer option, since we do not consider any attrition for the CWF. However, this results in lower IWF utilization. ADP does not discard high risk resources but rather has a higher utilization of the IWF resources. This practice can result in higher job reassignments due to IWF attrition coming into play. Hence ADP has better IWF utilization, higher levels of profitability, and more job reassignments.

The two-way analyses of variance conducted corroborates the results discussed above. It is also evident that ADP is better able to balance the two penalties than RH. While RH makes high use of CWF regardless of the reassignment penalty, ADP moderates its use based on the penalty level. ADP incurs higher idle penalty at the high level of reassignment penalty but the idle penalty incurred reduces at lower levels of reassignment penalty. This indicates that ADP intelligently decides against reassigning CWF jobs to the IWF when the penalty is high. When the reassignment penalty reduces, ADP brings back the CWF jobs to the IWF thus increasing IWF utilization.

7. SUMMARY AND CONCLUSIONS

7.1 Resource Planning under Uncertainty

The first objective of this research was to develop a model for resource planning in the service industry under the influence of uncertainty. With this aim in mind, we developed the MPSRP. The model contributes to the extant literature in several ways. First, it accounts for uncertainty in both resource and job availability. To the best of our knowledge, this is the first attempt at modeling uncertainty in both the supply and demand side of resource planning problems. Previous attempts at modeling multi-period resource planning either assumes the availabilities to be deterministic or considers partial uncertainty (either on the resource or job side). We also consider a complex staffing scenario where the potential set of jobs over the planning horizon is greater than the set of internal resources thus requiring the use of a contingent workforce. The CWF resources are less expensive than IWF resources but they also offer lower overall contribution.

This problem setting addresses the key issue of obtaining the appropriate resource mix which can be described as follows: when a service organization faces attrition among its internal resources, how does it create project staffing plans and to what extent does it need to depend on a contingent workforce to meet its demand? Another factor that makes the problem scenario more realistic is the prohibition of job reassignments due to the highly technical nature of projects that are being staffed. Job reassignments will tend to occur in order to balance IWF attrition and the goal here is to develop staffing plans that minimizes such job reassignments and dependency on CWF resources.

7.2 Stochastic Approximate Dynamic Programming Algorithm

The second goal of this research was to develop a tractable stochastic ADP algorithm for solving the MPSRP which is a complex combinatorial optimization problem. The exact dynamic programming algorithm is susceptible to the curses of dimensionality and is not suitable for solving real life problem sizes. ADP algorithms have been used intensively in recent years for overcoming the challenges faced by the exact DP solution methodology. The ADP algorithmic framework and the value function update procedures have been discussed in chapter 5.

The ADP algorithm is trained using a set of Monte Carlo samples over which it learns the impact of job and IWF resource uncertainty. The updated value function vectors capture both the impact of uncertainty and the contribution of each resource-job assignment. We develop a unique training mechanism that rewards optimal and feasible (in terms of availability) IWF resource-job assignments, and penalizes sub-optimal and infeasible IWF resource-job assignments. The value function vectors that result from the training phase is tested using a different set of Monte Carlo samples. The performance of the ADP algorithm is compared to that of a rolling horizon procedure, which is the commonly used approach to address multi-period problems.

Computational experiments has provided evidence that the ADP algorithm is advantageous over the RH procedure both in terms of solution quality and IWF utilization. A key objective of a service organization in determining its project staffing plan is maximizing its IWF utilization. The resource planning support provided by ADP makes higher utilization of IWF resources and generates more contribution from them when compared to the rolling horizon procedure. ADP's performance improvement over RH also becomes higher when the number of jobs to be staffed increases. That is, when the resource planning situation become complex, ADP outperforms RH to a greater extent.

The resource planning support provided by ADP makes maximal use of IWF resources, minimizes the dependency on CWF and generates higher profitability in the presence of resource and job uncertainty. ADP incurs a higher level of job reassignments but this is offset by the higher IWF utilization. This has a significant impact on the human resource recruiting policy and the need to develop the appropriate resource mix to satisfy probabilistic demand. Indeed, the intelligent balancing act provided by ADP to manage the reassignment and idle resource penalties offers appropriate levels of IWF and CWF job assignments under varying demand levels.

The ADP framework lends itself well to implementation in real life business setting. A graphical user interface (GUI) frontend can be added to the ADP framework to obtain a user friendly Stochastic Resource Planning (SRP) tool. Such a system would remove the user from the technical details of the algorithm. The users of such a system can be the HR operations manager, project team leaders and top management. The data that the user would need to run the tool would be the set of resources and jobs under consideration, the length of the planning horizon, the IWF attrition probabilities and the job win probabilities. The ADP training phase can be conducted in an offline setting. That is, using either estimated, historical or simulated data (IWF attrition probabilities and job win probabilities) the user can begin the training phase of the ADP algorithm. In the case that a simulated dataset is used, a Monte Carlo simulator can be built into the SRP tool. The addition of a simulator would provide the opportunity for the user to study different supply and demand patterns, in addition to the estimated and historical data at hand. The training phase can be run before the onset of the planning horizon.

Once the training phase is completed, its output (the updated VF vectors) can be input to the testing phase for resource planning. The testing phase can be conducted right before the start of each period of the planning horizon. The output of the testing phase will be a detailed resource plan that outlines the staffing requirements for the realized jobs in the current period. It will provide the specific mix of IWF and CWF required to staff all the jobs. Detailed information on the IWF resources who will be kept idle, job reassignments and the jobs outsourced to the CWF can be obtained.

7.3 Future Research

This research effort has laid the foundation for modeling multi-period resource planning in the presence of resource and job uncertainty. The MPSRP model and ADP algorithm has opened up possibilities of applying rigorous simulation based OR algorithms for solving this family of problems. Three lines of research related to this dissertation are possible in the future.

First, from a modeling perspective, there is a need to study the impact of CWF job assignments. That is, in our current model we do not consider CWF attrition or job reassignments between the CWF resources. While it is critical for a service organization to focus on IWF utilization, attrition and reassignment among the CWF will impact the contribution obtained from CWF assignments and will be worth investigating. It would also be useful to study variations in CWF contribution. That is, the CWF contribution might not always be positive. If the jobs are highly technical and require non-commodity skills (such as operations research, statistics, artificial intelligence), CWF resources might not be able to satisfactorily execute such jobs. Thus, it will be insightful to study the impact of zero or even negative CWF contributions. Another extension can be the assignment of a job to two or more resources which is quite reasonable and is encountered frequently in practice. Also, as discussed in chapter 2 it would be interesting to study the impact of FTE allocations on project staffing under uncertainty. Finally, in this research we assume the project win probabilities and IWF resource attrition probabilities to be static over the planning horizon. Modeling changes in the probabilities over the planning horizon would be a beneficial extension to this work.

Second, from an algorithmic perspective, there is room to develop training algorithms that exploit the problem structure and reduces computational time. As problem size increases, the current implementation of the ADP algorithm will become less desirable as it requires extensive effort for the training procedure. This is both an algorithmic issue and a modeling issue. There needs to be investigation into modeling the MPSRP into other forms such as network models, and also to modify the training mechanism such that it is more efficient. It would be beneficial to investigate the use of heuristics (for e.g., linear relaxation method) for the training phase as it is the most time consuming component of the ADP framework. The key issue here is obtaining and updating the value of each resource – job assignment pairs over the planning horizon. It is also critical to focus on methods to update the value of assignment pairs that have low probabilities and are less feasible.

Finally, from an application perspective, more real world applications can be modeled by the MPSRP. For example, our current formulation deals with project-oriented demand where jobs are decomposed from projects and reassignments are not desired. The model can be modified to deal with process-oriented demand like call centers where jobs are independent and are not project based. In this case it is possible to remove the reassignment constraint. In fact, job reassignments will be encouraged in such a case with multi-skilled resource. It is a natural extension of this research and will make the MPSRP more generalizable. A different set of computational experiments that vary the point estimate thresholds based on the decision maker's risk profile will be beneficial. The point thresholds are used by the decision maker to fix future resource and job availabilities for the deterministic RH procedure. It would be insightful to investigate the impact of different thresholds for resource and job availabilities. Another extension is the inclusion of project scheduling to the resource planning support. For example, the jobs that make up a project might need to be executed in phases due to dependencies. Our current assumption is that all the jobs of a projects can be executed in parallel as soon as they are won. There are cases where job 1 of a project need to be executed before work on job 2 can start, and so on. This is an important theoretical consideration that should be investigated.

LIST OF REFERENCES

Ahuja, R. K., Kumar, A., Jha, K. C., & Orlin, J. B. (2007). Exact and heuristic algorithms for the weapon-target assignment problem. *Operations research, 55*(6), 1136-1146.

Albareda-Sambola, M., Fernández, E., & Saldanha-da-Gama, F. (2011). The facility location problem with Bernoulli demands. *Omega, 39*(3), 335-345.

Albareda-Sambola, M., van der Vlerk, M. H., & Fernández, E. (2006). Exact solutions to a class of stochastic generalized assignment problems. *European Journal of Operational Research, 173*(2), 465-487.

Albright, C. (1974). A Markov-decision-chain approach to a stochastic assignment problem. *Operations research, 22*(1), 61-64.

Albright, C. (1977). A Bayesian approach to a generalized house selling problem. *Management Science, 24*(4), 432-440.

Albright, C., & Derman, C. (1972). Asymptotic optimal policies for the stochastic sequential assignment problem. *Management Science, 19*(1), 46-51.

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America, 38*(8), 716.

Bertsekas, D. P., & Castanon, D. A. (1999). Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics, 5*(1), 89-108.

Bertsekas, D. P., Tsitsiklis, J. N., & Wu, C. (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics, 3*(3), 245-262.

Çalışkan, C., & Hall, R. W. (2003). A dynamic empty equipment and crew allocation model for long-haul networks. *Transportation Research Part A: Policy and Practice, 37*(5), 405-418.

Cattrysse, D. G., & Van Wassenhove, L. N. (1992). A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research, 60*(3), 260-272.

Chen, Z. L., & Xu, H. (2006). Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science, 40*(1), 74-88.

Chow, Y., Moriguti, S., Robbins, H., & Samuels, S. (1964). Optimal selection based on relative rank (the "secretary problem"). *Israel Journal of Mathematics*, *2*(2), 81-90.

Chun, Y. H., Moskowitz, H., & Plante, R. D. (1994). Dynamic programming formulation of the group interview problem with a general utility function. *European Journal of Operational Research, 78*(1), 81-92.

Chun, Y. H., & Sumichrast, R. T. (2006). A rank-based approach to the sequential selection and assignment problem. *European Journal of Operational Research, 174*(2), 1338-1344.

David, I., & Yechiali, U. (1995). One-attribute sequential assignment match processes in discrete time. *Operations research, 43*(5), 879-884.

Derman, C., Lieberman, G. J., & Ross, S. M. (1972). A sequential stochastic assignment problem. *Management Science*, 349-355.

Derman, C., Lieberman, G. J., & Ross, S. M. (1975). A stochastic sequential allocation model. *Operations research*, *23*(6), 1120-1130.

Drexl, A. (1991). Scheduling of project networks by job assignment. *Management Science, 37*(12), 1590-1602.

Dupont, A., Linhares, A. C., Artigues, C., Feillet, D., Michelon, P., & Vasquez, M. (2009). The dynamic frequency assignment problem. *European Journal of Operational Research, 195*(1), 75-88.

Ferguson, A. R., & Dantzig, G. B. (1956). The allocation of aircraft to routes—An example of linear programming under uncertain demand. *Management Science, 3*(1), 45-73.

Foulds, L., & Wilson, J. M. (1999). On an assignment problem with side constraints. *Computers & Industrial Engineering, 37*(4), 847-858.

Gass, S. I., Collins, R. W., Meinhardt, C. W., Lemon, D. M., & Gillette, M. D. (1988). OR Practice—The Army Manpower Long-Range Planning System. *Operations research, 36*(1), 5-17.

Goel, A. (2009). Vehicle scheduling and routing with drivers' working hours. *Transportation Science, 43*(1), 17-26.

Gosavi, A. (2003). *Simulation-based optimization: parametric optimization techniques and reinforcement learning* (Vol. 25): Springer.

Haghani, A., & Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research, 32*(11), 2959-2986.

Holder, A. (2005). Navy personnel planning and the optimal partition. *Operations research*, 77-89.

Kennedy, D. (1986). Optimal sequential assignment. *Mathematics of Operations Research, 11*(4), 619-626.

Kennington, J., & Wang, Z. (1992). A shortest augmenting path algorithm for the semi-assignment problem. *Operations research, 40*(1), 178-187.

Kleywegt, A. J., & Papastavrou, J. D. (1998). The dynamic and stochastic knapsack problem. *Operations research, 46*(1), 17-35.

Kogan, K., Khmelnitsky, E., & Ibaraki, T. (2005). Dynamic Generalized Assignment Problems with Stochastic Demands and Multiple Agent--Task Relationships. *Journal of Global Optimization, 31*(1), 17-43.

Kogan, K., & Shtub, A. (1997). DGAP-the dynamic generalized assignment problem. *Annals of Operations Research, 69*, 227-239.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1-2), 83-97.

Lacity, M. C., Solomon, S., Yan, A., & Willcocks, L. P. (2011). Business process outsourcing studies: a critical review and research directions. *Journal of information technology, 26*(4), 221-258.

Lennon, C., McGowan, J., & Lin, K. (2007). A game-theoretic model for repeated assignment problem between two selfish agents. *Journal of the Operational Research Society, 59*(12), 1652-1658.

Li, H., & Womer, K. (2009). A decomposition approach for shipboard manpower scheduling. *Military Operations Research*, *14*(3), 67-90.

McLay, L. A., Lee, A. J., & Jacobson, S. H. (2010). Risk-based policies for airport security checkpoint screening. *Transportation Science*, *44*(3), 333-349.

Meisel, S., Suppa, U., & Mattfeld, D. (2011). Serving Multiple Urban Areas with Stochastic Customer Requests. *Dynamics in Logistics*, 59-68.

Nakai, T. (1986). A sequential stochastic assignment problem in a partially observable Markov chain. *Mathematics of Operations Research*, *11*(2), 230-240.

Nikolaev, A. G., Jacobson, S. H., & McLay, L. A. (2007). A sequential stochastic security system design problem for aviation security. *Transportation Science, 41*(2), 182-194.

Papastavrou, J. D., Rajagopalan, S., & Kleywegt, A. J. (1996). The dynamic and stochastic knapsack problem with deadlines. *Management Science, 42*(12), 1706-1718.

Powell, W. B. (1996). A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science, 30*(3), 195-219.

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality* (Vol. 703): Wiley-Blackwell.

Powell, W. B., Bouzaiene-Ayari, B., Berger, J., Boukhtouta, A., & George, A. P. (2011). The Effect of Robust Decisions on the Cost of Uncertainty in Military Airlift Operations. *ACM Transactions on Modeling and Computer Simulation (TOMACS), 22*(1), 1.

Powell, W. B., & Carvalho, T. A. (1997). Dynamic control of multicommodity fleet management problems. *European Journal of Operational Research, 98*(3), 522-541.

Powell, W. B., & Carvalho, T. A. (1998). Dynamic control of logistics queueing networks for large-scale fleet management. *Transportation Science, 32*(2), 90-109.

Powell, W. B., Carvalho, T. A., Godfrey, G. A., & Simão, H. P. (1995). Dynamic fleet management as a logistics queueing network. *Annals of Operations Research, 61*(1), 165-188.

Powell, W. B., Snow, W., & Cheung, R. K. (2000). Adaptive labeling algorithms for the dynamic assignment problem. *Transportation Science, 34*(1), 50-66.

Powell, W. B., & Topaloglu, H. (2006). Approximate dynamic programming for large-scale resource allocation problems. *TutORials in Operations Research, M. Johnson, B. Norman, and N. Secomandi, Eds*.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*: John Wiley & Sons, Inc.

Righter, R. (1989). A resource allocation problem in a random environment. *Operations research, 37*(2), 329-338.

Righter, R. (2011). Stochastic sequential assignment problem with arrivals. *Probability in the Engineering and Informational Sciences, 25*(4), 477.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 400-407.

Rose, J. S. (1982). A problem of optimal choice and assignment. *Operations research, 30*(1), 172-181.

Ross, S. M., & Wu, D. T. (2012). A generalized coupon collecting model as a parsimonious optimal stochastic assignment model. *Annals of Operations Research*, 1-14.

Saaty, T. L. (1990). How To Make A Decision: The Analytic Hierarchy Process. *European Journal of Operational Research, 48*(1), 9-26.

Santos, C., Gonzalez, T., Li, H., Chen, K.-Y., Beyer, D., Biligi, S., . . . Ramanujam, R. (2013). HP Enterprise Services uses optimization for resource planning. *Interfaces, 43*(2), 152-169.

Sennott, L. I., Van Oyen, M. P., & Iravani, S. M. R. (2006). Optimal dynamic assignment of a flexible worker on an open production line with specialists. *European Journal of Operational Research, 170*(2), 541-566.

Sethi, S., & Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research, 29*(1), 387-415.

Shestak, V., Chong, E. K. P., Maciejewski, A. A., & Siegel, H. J. (2009). *Robust sequential resource allocation in heterogeneous distributed systems with random compute node failures.* Paper presented at the Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on.

Shestak, V., Chong, E. K. P., Maciejewski, A. A., & Siegel, H. J. (2012). Probabilistic resource allocation in heterogeneous distributed systems with random failures. *Journal of Parallel and Distributed Computing*.

Su, X., & Zenios, S. (2002). Allocation of kidneys to autonomous transplant candidates: A sequential stochastic assignment model. *Oper. Res*.

Su, X., & Zenios, S. A. (2005). Patient choice in kidney allocation: A sequential stochastic assignment model. *Operations research*, 443-455.

Tadei, R., & Ricciardi, N. (1999). The dynamic multilevel assignment problem as a stochastic extremal process. *European Journal of Operational Research, 117*(2), 264-274.

Terrab, M., & Odoni, A. R. (1993). Strategic flow management for air traffic control. *Operations research, 41*(1), 138-152.

Thomas, B. W., & White, C. C. (2007). The dynamic shortest path problem with anticipation. *European Journal of Operational Research, 176*(2), 836-854.

Topaloglu, H., & Powell, W. B. (2006). Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS Journal on Computing, 18*(1), 31-42.

Turner, J. P., Lee, S., Daskin, M. S., Homem-de-Mello, T., & Smilowitz, K. (2009). Dynamic fleet scheduling with uncertain demand and customer flexibility. *Computational Management Science*, 1-23.

Wang, F., Yang, M., & Yang, R. (2006). *Dynamic fleet management for Cybercars.* Paper presented at the Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE.

Wardrop, J. (1952). Some Theoretical Aspects of Road Traffic Research, reprinted in H. Mohring eds. *The Economics of Transoport I*, 299-352.

Come, Thou fount of every blessing,
tune my heart to sing Thy grace.
Streams of mercy, never ceasing,
call for songs of loudest praise.
Teach me some melodious sonnet,
sung by flaming tongues above;
Praise His name, I'm fixed upon it,
name of God's redeeming love.

Hitherto, Thy love has blessed me,
Thou hast drawn me to this place.
And I know Thy hand will lead me,
safely home by Thy good grace.
Jesus sought me when a stranger,
wandering from the fold of God;
He to rescue me from danger,
bought me with His precious blood.

O to grace, how great a debtor,
daily I'm constrained to be.
Let Thy goodness, like a fetter,
bind my wandering heart to Thee.
Prone to wander, Lord I feel it,
prone to leave the God I love;
Here's my heart, O take and seal it,
seal it for Thy courts above.