A Reinforcement Learning Approach

To Obtain Treatment Strategies In Sequential Medical Decision Problems

by

Radhika Poolla

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Industrial Engineering
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Major Professor: Tapas K. Das, Ph.D.
Jose L. Zayas-Castro, Ph.D.
Deepak K. Agrawal, Ph.D.

Date of Approval:
August 14, 2003

Keywords: dynamic decision model, markov decision process, hereditory spherocytosis, intervention, quality adjusted life years, average reward

ACKNOWLEDGEMENTS

I would like to express my grateful thanks for the help and advice given by my major professor, Dr. Tapas K. Das, who has been the inspiration for many students and many more to come. He is a teacher by example, a great mentor in influencing his students, and a great friend in his interaction. It is a great experience working with him and I look forward to learn many more things, with a continued interaction in future.

I owe my sincere thanks to Dr. Jose L. Zayas-Castro, for all his interest in my work, for his encouragement and support and for giving many valuable comments on the manuscript. I would like to give special thanks to Dr. Deepak Agrawal, for accepting to be on my committee, for his valuable suggestions on the problem and for being very cooperative.

I thank our engineer, Chris Paulus, program assistant, Gloria Hanshaw, office manager Jackie Stephens and Marsha Brett for all their help at the USF.

I would like to express my thanks to my friend, guide and also, senior at the graduate school, Kiran Ravulapati for motivating me in various ways, for his discussions on the work, for all those great trips to Atlanta, and for all the good times shared during our graduate years. I would like to make a special note of my friend Srinivas Kalla, for all the care, great guidance, encouragement and support, all these years of my graduate studies at Tampa and look forward to his great company many more years ahead. A hearty thanks goes to my best friends, Pawan Katharikuppam and Sridhar Mohan for all the great times shared, for all the fun and most of all, for being very dependable.

TABLE OF CONTENTS

LIST OF TABLES iv

LIST OF FIGURES v

ABSTRACT vii

CHAPTER 1 INTRODUCTION 1
- 1.1 Sequential decision problems 2
- 1.2 Some medical decision problems 3
  - 1.2.1 Spontaneous pnemothorax 3
  - 1.2.2 Chronic angina (chest pain) 4
  - 1.2.3 Chronic cough 4
  - 1.2.4 Severe head injury management 4
  - 1.2.5 Colorectal cancer follow up 5
  - 1.2.6 Chronic leukemia 5
- 1.3 Current approaches 6
  - 1.3.1 Static models 6
  - 1.3.2 MDP & SMDP 6
  - 1.3.3 Graphical formalisms 7
    - 1.3.3.1 Dynamic influence diagrams 7
    - 1.3.3.2 Markov cycle trees 9
    - 1.3.3.3 State transition diagrams 10
    - 1.3.3.4 Influence views 10
    - 1.3.3.5 Decision trees 11
  - 1.3.4 Neural networks 12
  - 1.3.5 Belief networks 12
  - 1.3.6 Genetic algorithms 13
  - 1.3.7 Rough set theory 13
- 1.4 Brief description of the problem 13
- 1.5 Existing solution methodology 14
- 1.6 Need for better methods 15
- 1.7 Approach considered 16
  - 1.7.1 Reinforcement learning (RL) 16
- 1.8 Summary of remaining chapters 17

CHAPTER 2 LITERATURE REVIEW 18
2.1 Decision trees 18
2.2 Markov cycle trees 19
2.3 Stochastic trees 20
2.4 Markov models 21
2.5 Dynamic decision models 22
2.6 Obtaining the numbers 28
2.7 Static modeling 29

CHAPTER 3 RESEARCH OBJECTIVES 32
3.1 Problem statement 32
3.2 Research objectives 33

CHAPTER 4 PROBLEM FORMULATION AND SOLUTION METHODOLOGY 34
4.1 Problem formulation 34
4.1.1 Elements of the MDP 36
4.1.1.1 State space 36
4.1.1.2 Action space 36
4.1.1.3 Time horizon 37
4.1.1.4 Decision epoch 37
4.1.1.5 Transition probabilities 37
4.1.1.6 Rewards 38
4.1.2 Quality adjusted life years (QALY) 38
4.1.2.1 Utility function 38
4.1.2.2 QALY 39
4.1.2.3 Methods for deriving quality weights for health states 41
4.1.2.4 Rating scale 42
4.1.2.5 Standard gamble 43
4.1.2.6 Time tradeoff 44
4.1.2.7 Multi-attribute health status surveys 45
4.1.2.8 Cost-utility ratios 46
4.1.2.9 Limitations of QALYs 47
4.1.2.10 Uses of QALYs 48
4.1.2.11 Method followed to derive quality weights for health states 48
4.1.2.12 Immediate rewards in terms of QALYs 51
4.1.3 Hereditory spherocytosis 52
4.1.3.1 Spleen 52
4.1.3.2 Gallstones 53
4.1.3.3 Sepsis 54
4.1.3.4 Time 55
4.1.3.5 Complications 55
4.1.3.6 Age 57

4.1.3.7 Sex 58
4.2 Model solution 58
4.2.1 Simulation mechanism 58
4.2.1.1 Assignment of starting state 59
4.2.1.2 Input parameters 59
4.2.2 Average reward reinforcement learning 60
4.2.2.1 RL algorithm 60

CHAPTER 5 NUMERICAL RESULTS 63
5.1 Reinforcement methodology results 63
5.2 Value iteration approach 66
5.2.1 Method to obtain transition probability matrices (TPMs) 66
5.2.2 Method followed to obtain reward matrix 70
5.3 Policy differences 72

CHAPTER 6 CONCLUSIONS 74
6.1 Concluding remarks 74
6.2 Extensions to this work 76

REFERENCES 77

APPENDICES 82
Appendix A MARKOV DECISION PROCESS 83
A.1 Bellman optimality equation for average reward MDP's 85
A.2 The average reward value iteration algorithm 85

Appendix B REINFORCEMENT LEARNING 88
B.1 Average reward RL 91
B.2 Model based RL 92
B.3 Model free RL 92
B.4 RL and DP 93
B.5 RL and temporal difference methods 94

LIST OF TABLES

Table 1. Quality weights 49

Table 2. Results from the RL methodology 64

Table 3. State-variable transition probabilities in a decision epoch 67

Table 4. Differences in policies of value iteration and reinforcement learning 73

LIST OF FIGURES

Figure 1. Dynamic influence diagram 8

Figure 2. Markov cycle tree 9

Figure 3. State transition diagram 10

Figure 4. Influence view 11

Figure 5. Rating scale for quality weights 43

Figure 6. Standard gamble for deriving quality weights 44

Figure 7. State transition diagram for gallstones 54

Figure 8. Average reward values for different exploration parameter values 65

Figure 9. A reinforcement learning model 89

A REINFORCEMENT LEARNING APPROACH

TO OBTAIN INTERVENTION STRATEGIES IN MEDICINE

Radhika Poolla

ABSTRACT

Medical decision problems are extremely complex owing to their dynamic nature, large number of variable factors, and the associated uncertainty. Decision support technology entered the medical field long after other areas such as the airline industry and the manufacturing industry. Yet, it is rapidly becoming an indispensable tool in medical decision making problems including the class of sequential decision problems. In these problems, physicians decide on a treatment plan that optimizes a benefit measure such as the treatment cost, and the quality of life of the patient. The last decade saw the emergence of many decision support applications in medicine. However, the existing models have limited applications to decision problems with very few states and actions. An urgent need is being felt by the medical research community to expand the applications to more complex dynamic problems with large state and action spaces. This thesis proposes a methodology which models the class of sequential medical decision problems as a Markov decision process, and solves the model using a simulation based reinforcement learning (RL) algorithm. Such a methodology is capable of obtaining near

optimal treatment strategies for problems with large state and action spaces. This methodology overcomes, to a large extent, the computational complexity of the value-iteration and policy-iteration algorithms of dynamic programming. An average reward reinforcement-learning algorithm is developed. The algorithm is applied on a sample problem of treating hereditary spherocytosis. The application demonstrates the ability of the proposed methodology to obtain effective treatment strategies for sequential medical decision problems.

CHAPTER 1

INTRODUCTION

Ability to reason differentiates humans from other species. Reasoning leads humans to perceive, understand, analyze, and act. Humans act by making decisions and this process happens almost every minute of our lives. In situations involving many variables and possible decisions, decision support systems provide useful tools. A decision support system translates the real life scenario into a mathematical model for analysis. A set of decisions usually evolves from this process and, generally, the decision that best satisfies the objective of the analysis is carried out.

Decision support systems have been gaining usage in many application areas including, pharmacy, manufacturing, finance, armed forces, aviation industry, and health sciences. Because of the complexity of decision making, health sciences have been a new and fast growing field of application. Factors such as multiple variables, uncertainty of action outcomes, difficulty of incorporating input obtained from domain experts into the model building process, and the time varying nature of the problems pose a tough challenge to the decision support experts as they try to fit such complexities into mathematical frameworks, which are more parameterized. Techniques from the fields of Statistics and Probability are proving useful to model some of these complex situations efficiently and to arrive at the best possible decisions.

1.1 Sequential decision problems

Diagnostic testing, therapy planning, and other clinical scenario, comprise of the physical condition of the patients, the interventions, which are diagnostic tests and treatments, or a combination of both. These, medical scenarios, usually, comprise of problems which, involve a trade-off between certain events affecting the health of a patient and the risk of a certain intervention to avert the events. Both the associated risk and the health of a patient may vary over time, which makes the situation uncertain for the physician to predict accurately. The objective of such medical problems is to find a suitable therapeutic plan for the patient under observation, which would maximize the quality of life of the patient in a cost effective manner.

A typical sequential decision problem arises when a patient approaches the physician, and the physician, depending upon the patient's health situation, decides to either intervene immediately or to wait and see for some time, with the objective of maximizing the quality of life for the patient. If the physician believes that the patient's life is at risk or the patient's health would be severely affected if he or she were left in the same condition, the physician might opt for an intervention. But if the physician is unsure about the need for an intervention and prefers to keep the patient under observation, then, a preferred strategy could be 'wait and see'. Questions listed below could arise in the case of adopting a 'wait and see' strategy.

- How long should the physician observe the patient before the decision is revisited?
- Should the patient's condition be continuously monitored or in discrete intervals?

In the case of interventions, the side effects from the interventions can lead the patient into a different situation, which the physician may not be able to predict with certainty. Moreover, there could be many modes of interventions, such as medicinal and surgical. Selecting a mode that would provide the best possible treatment to the patient at that particular time and situation could be a difficult task.

Age of the patient and sex might be two other factors, which the physician has to keep in mind, while taking such a decision. Ethnicity of the patient may not be taken into consideration. In addition to all these, another problem feature, which confronts the physician is the dynamicity of the problem. A patient's physical condition may vary with time during the course of the treatment. For such problems, decision support systems could help the physicians in taking quick and efficient decisions to maximize the quality adjusted life years (QALY) of a patient in the long run. A QALY is a measure of the quality life that the patient enjoys in a year.

1.2 Some medical decision problems

1.2.1 Spontaneous pnemothorax

The problem of finding an optimal strategy for primary spontaneous pneumothorax, (Lin et al. (2002) [1]), in young men is a typical decision problem, that falls under the category of intervention problems. This has been modeled using a Markov decision process with a state space of five and an action space of six. The objective was to maximize the quality adjusted life years of a patient.

1.2.2 Chronic angina (chest pain)

In the case of chronic stable angina, the decision problem involved is to determine the treatment and the time of treatment such that the quality adjusted life expectancy of a patient is optimized. The actions usually available in this scenario are medical treatments, percutaneous transluminal angioplasty, and coronary artery bypass graft. While the selected treatment progresses, complications occur requiring other decisions. Hence, the sequence of decisions taken depending on the situation of a patient is very crucial to maximize the objective function, the quality-adjusted life expectancy. This problem was modeled in the literature as a Markov decision process (MDP) having five state variables and three actions (Leong T.Y. (1994) [2]).

1.2.3 Chronic cough

This problem, (Lin et al. (2002) [3]), involves, finding the most cost-effective management strategy, out of the available strategies, to treat chronic unexplained cough. The model used is a MDP with six treatment strategies.

1.2.4 Severe head injury management

In the case of severe head injury, (Harmanec et al. (1999) [4]), the management becomes extremely difficult owing to the time-critical nature of the injury, the complexities involved in the scenario, and the uncertainty of the intervention procedures. The decision model presented in [4] considers nine treatment options and the influence diagram approach.

1.2.5 Colorectal cancer follow up

Patients with colorectal cancer undergo curative surgery. The follow up period after the surgery is very important as there could be either recurrence of the cancer or development of tumor or both. If the recurrence or tumor is detected at an early stage during the follow-up, the chance of successful curative treatment can be improved. For the detection, the doctor needs to perform a series of diagnostic tests.

The decision problem (Zheng et al. (1998) [5]) here, is to find out the optimal course of tests depending on the stage of health of the patient during the follow-up, which would ultimately lead to the most cost-effective treatment sequence. This problem was modeled with seven actions and five state variables as a semi-markov decision process (SMDP). This model has been solved using the value-iteration technique by using DynaMol- dynamic decision modeling language, developed by T.Y. Leong (1998) [32], which takes inputs as the conditional probabilities and the influence view of the problem.

1.2.6 Chronic leukemia

Patients who are born with errors in their immune system and patients who have diseases like severe aplastic anaemia, and chronic leukemia are treated by allogenic bone marrow transplantation. But during this transplantation, the patient's cells could develop a negative reaction to the donor's cells. This complication is called graft versus host disease (GVHD), which occurs frequently and is deadly.

In the case of leukemia patients, mild GVHD helps in preventing disease relapse. Therefore, though severe GVHD is dangerous, mild form of GVHD is advantageous to the transplantation. Leukemia patients are treated with immuno suppressive drugs in

order to prevent or control GVHD. The dosage of these drugs should be optimal such that they clear the complication caused by GVHD and at the same time control the GVHD to benefit the transplantation.

Thus, the decision problem (Paolo Magni et al. (1997) [6]) is to specify both type and dosage of the drugs in order to either avoid or to induce GVHD according to the patient's specific condition and drug's toxicity. This problem has been modeled as a MDP with four actions and five state variables forming the state space. Influence views were used to model the problem. The details were supplied to a software called DT-Planner, which models the problem as an MDP and solves for an optimal policy using well known algorithms, such as value iteration and policy iteration. The policy that maximizes the survival time while minimizing the risk of drug toxicity was adopted.

1.3 Current approaches

The main approaches that have been used in studying the problems discussed above are given below.

1.3.1 Static models

In this approach, the decision problems are solved at several time instants and the set of solutions are then presented as a dynamic strategy. Such a model presents a crude approximation and leads to a sub-optimal solution.

1.3.2 MDP & SMDP

A Markov Decision Process (MDP) model consists of a set of possible states S, a set of possible actions A, a reward function R(s, a). The actions can be of two types, namely, deterministic and stochastic. Deterministic actions are those, where, for each

state and action, a particular new state is defined. Where as, for a stochastic action, for each state and action, a probability distribution has to be specified over the next states.

The solution expected to a problem, by modeling as a Markov decision process is an optimal policy. Optimal policy tells, which action to be followed in a particular state, so that, the total expected reward could be maximized. Semi-Markov Decision Process (SMDP) goes a step further in taking the time spent in a particular state also, into account for analysis. Medical problems can fit into these mathematical models, though with some assumptions and constraints.

1.3.3 Graphical formalisms

Many decision-making frameworks make use of graphical formalisms to easily accommodate the complexities of the problems. These formalisms by themselves cannot give a solution to a problem. They have an underlying mathematical framework, which models the actual problem. These formalisms, as given below, are useful for easy understanding of the problem. Below given are some of the formalisms in use.

1.3.3.1 Dynamic influence diagrams

Dynaimic influence diagrams are direct acyclic graphs. T.Y. Leong [32] depicted a influence duagram, which is as shown in Figure 1. The squares denote the decision nodes, the circles denote the chance nodes and the rhombus' denote the value nodes. Inside, each node, there is a number, which indicates the decision stage in which the decision/event/value is considered. Arcs leading to chance and value nodes in the figure denote the probabilistic dependencies and arcs leading to the decision nodes indicate the informational dependencies. The possible value of the outcome of a chance node or a

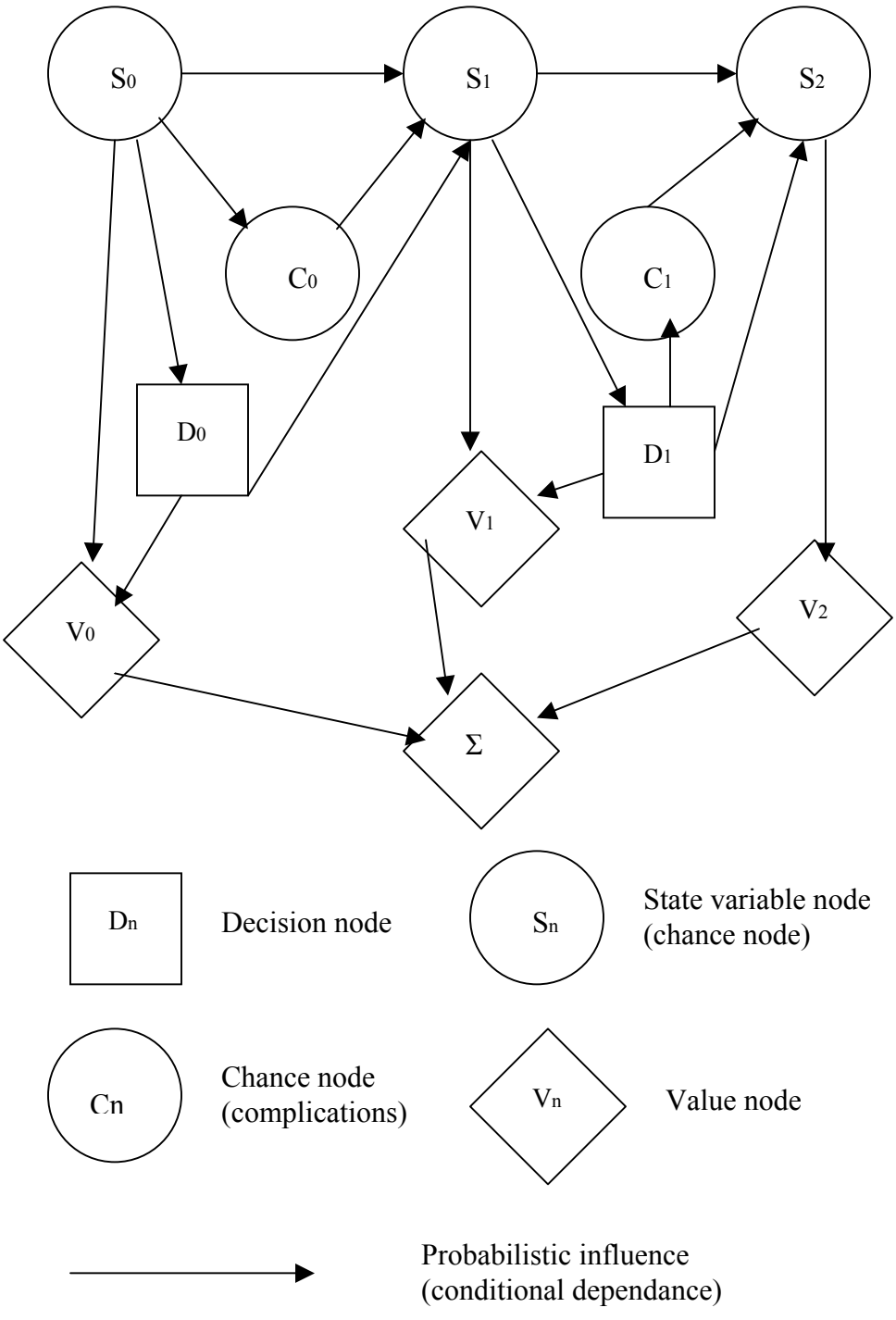

Figure 1. Dynamic influence diagram

value node is embedded in each of them. One diagram is enough to model a situation with any number of actions.

1.3.3.2 Markov cycle trees

In a Markov cycle tree, the branches of the tree come out of the root node, which is called as the Markov node. For a given action, the leaf nodes represent the states at the beginning and at the end of a decision stage.

The arcs indicate the possible outcomes and also the conditional dependencies among the nodes. A utility function is always defined for each of the states in the diagram. The number of Markov cycle trees for a given problem will be equal to the number of actions available. The uncertainty in the problem and the variation with time would lead to extreme complexity of the Markov cycle trees. T.Y. Leong depicted the Markov cycle tree in [32], as shown in Figure 2.

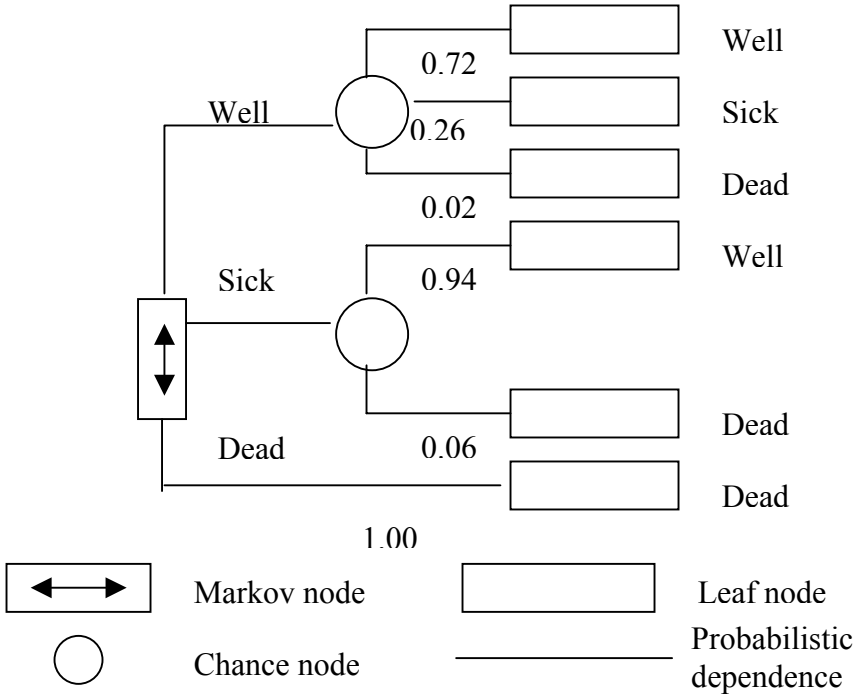

Figure 2. Markov cycle tree

1.3.3.3 State transition diagrams

As shown in Figure 3, the nodes denote the states and the arcs denote the possible transitions given an action. The transition probabilities are denoted above the arcs. A utility function is defined for each state in the diagram.

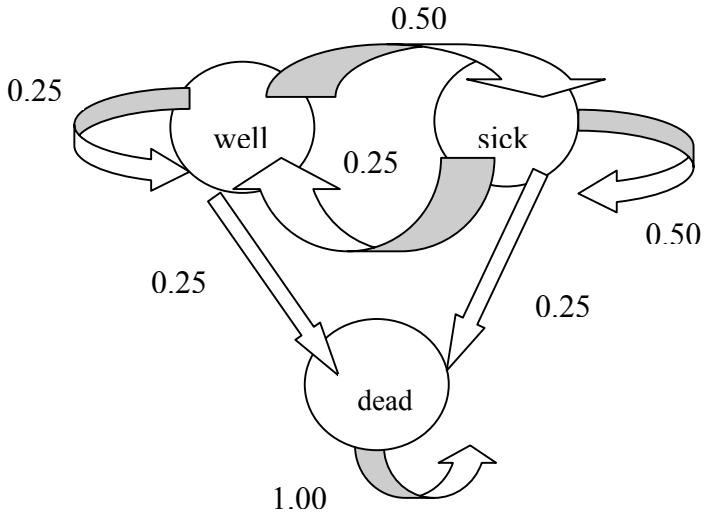

Figure 3. State transition diagram

1.3.3.4 Influence views

An influence view is a diagram wherein the events taking place in a single transition are modeled. For each action defined in the problem, an influence view can be drawn. This is very similar to the transition diagram, except that, in this, the events are modeled as nodes whereas in a transition diagram, the states are modeled as nodes. Also, in an influence view, a conditional distribution table is associated with each node, which is comparable to the transition probabilities associated with the arcs in a transition diagram, only difference being that the transition probabilities are far more difficult to obtain than the conditional probabilities. Paolo Magni et al. [8] depict an influence view

as shown in Figure 4. The information obtained from an influence view can always be obtained from a properly drawn transition view, except for the difficulty of obtaining the exact transition probabilities from the existing medical databases concerning the problem.

Artificial Intelligence researchers have been working on the dynamic decision problems with other methodologies, like the ones mentioned below. Sometimes, statistical techniques and AI methods are being combined and used for modeling.

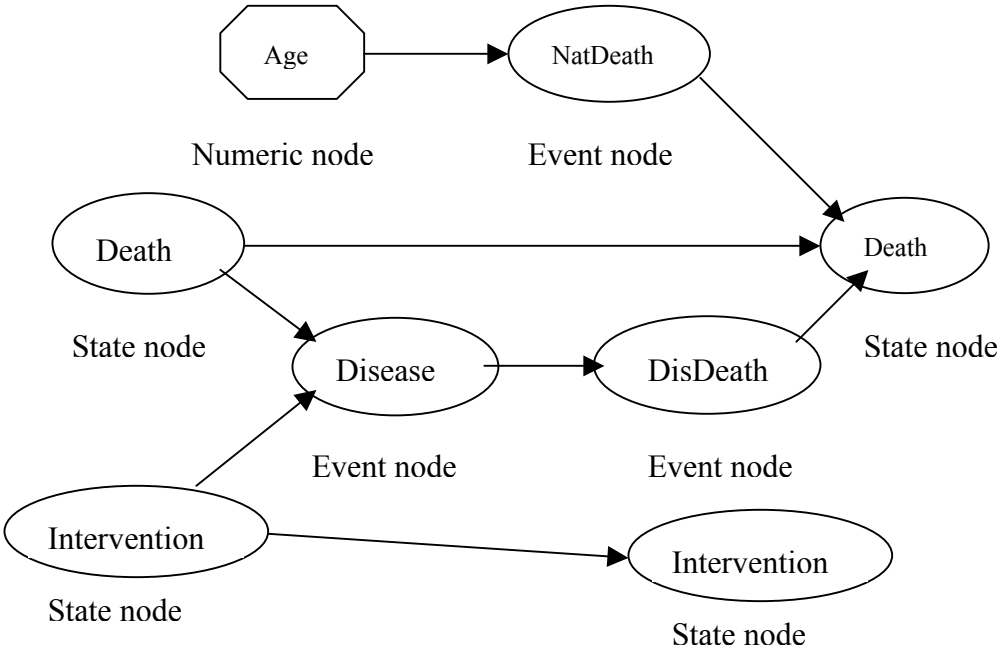

Figure 4. Influence view

1.3.3.5 Decision trees

Decision trees have always been popular in sequential decision-making. The other advantages of a decision tree are that, it can easily be translated into convenient if-then rules. Constraints also can be easily imposed. However, the decision tree needs to be

learned through heuristic procedures, as the problem of finding the best tree is an NP-hard problem. The major disadvantage of decision trees is that they are not suitable for time varying problems.

1.3.4 Neural networks

A neural network consists of a set of nodes called the input nodes, output nodes and intermediate nodes. Input nodes receive the input signals. Output nodes give the output signals and a large number of intermediate layers contain the intermediate nodes. Such networks can be built using special hardware, but most of them are just software programs that can operate on normal computers.

There are two stages involved in the neural network learning,

- Encoding stage: Neural network is trained to perform a certain task,
- Decoding stage: Neural network is used to classify examples, make predictions or execute whatever learning task is involved.

Different forms of neural networks are perceptrons, back propagation networks, and kohonen self-organizing map.

1.3.5 Belief networks

Belief networks help in modeling phenomenon, which have an uncertainty element. They deal with reasoning under uncertainty. Bayesian belief networks are directed acyclic graphs with a set of nodes interconnected with arcs. Each node represents an uncertain quantity or a random variable. The arcs link the variables, which have direct influence over each other. The influences are shown over the arc with the help of

conditional probabilities. Belief networks have applications in medical diagnostic systems, weapons scheduling, and computer processor fault diagnosis, to name a few.

1.3.6 Genetic algorithms

These are basically adaptive, heuristic search algorithms based on the evolutionary ideas of Charles Darwin. Their intelligent exploitation of a random search within a defined search space to solve a problem makes them outperform other traditional methods. Being good at solving problems, involving, finding optimal parameters, they are especially useful in optimization.

Genetic algorithms can be applied to problems where the search space is large and complex, domain knowledge is scarce and where mathematical analysis is not available. Machine and robot learning, economic models, ecological models and automated programming are some of the areas, for which, genetic algorithms have been applied.

1.3.7 Rough set theory

Rough set theory mainly deals with classification of data tables. It is one of the techniques available to search large databases for meaningful decision rules and to acquire new knowledge. It has found applications in medical data analysis, image processing and voice recognition.

1.4 Brief description of the problem

One such decisio problem is the Hereditary Spherocytosis problem considered in this thesis. In this disease the patient suffers from being anemic because of the red blood cell destruction. If the patient is not cured, then there is an increasing risk of gall stone formation, in addition to the red-blood cell destruction. On the other hand, if the

physician intervenes, in an attempt to cure the patient, a septic condition called sepsis can develop.

Five possible interventions are available for the physician to choose from, depending upon the patient's condition. But, the problem lies in taking these decisions at appropriate patient conditions so as to maximize the quality adjusted life days of the patient. The patient's condition changes continuously adding a dynamic dimension to the problem. The changing condition of the patient, the side effects arising from the medical interventions, and the amount of patient discomfort are some of the issues that a physician has to continuously monitor and keep in mind while choosing the intervention strategy.

1.5 Existing solution methodology

The problem of selecting an intervention strategy for Hereditory Spherocytosis has been modeled in the literature using a static modeling formalism by Marchetti et al. (1998) [7]. Later, it has also been modeled by Paolo Magni et al. (2000) [8] as a Markov decision process to accomodate the dynamic perspective. The Markov cycle has been fixed at one year. Influence views were used to describe the effects of the four possible action choices. State of gallstones and state of spleen characterized a patient's health condition or states of the MDP. Quality adjusted life years is considered as the utility function. The decision problem is to find the best action in every state of the patient to maximize the quality adjusted life expectancy of a patient. Obtaining transition probabilities for the states for every action and then solving for the optimal policy using the existing value iteration algorithm constitute the solution procedure of the Markov

decision process. The transition probabilities are usually deduced from the conditional probabilities obtained from the medical databases.

1.6 Need for better methods

There are two existing dynamic programming algorithms to solve for the optimal policy of a MDP, namey, value and policy iteration. The computational complexity of the value-iteration algorithm per iteration is quadratic in the number of states and linear in the number of actions. In other words, each iteration can be performed in $O(|A| |S|^2)$ steps. On the other hand, policy iteration converges faster than value iteration, but takes $O(|A| |S|^2 + |S|^3)$ steps per iteration. Thus, the computational complexity increases enormously with even a slightest increase in the action and state spaces.

Most of the medical problems, when modeled as a MDP or as a SMDP, because of the very nature of the problems, could end up with a large state space and a number of possible actions. For such problems, it becomes difficult to arrive at the optimal policy because of the issue of the computational complexity. The transition probability matrices become very large requiring lot of memory to store all the states. Also, much computational time is required for the value iteration or the policy iteration algorithms to converge, which is not feasible. Therefore, computationally efficient approaches are needed to obtain the optimal policy.

In the models studied in the literature, the state space of the Hereditory Spherocytosis problem has been reduced considerably comprising of only the state of gallstones and the state of spleen. Age and sex have not been considered. Moreover, time after splenectomy, sepsis formation, and other complexities have all been ignored in

establishing the state space. Thus, even though the problem has been studied as a MDP, significant elements of the problem have been left out to achieve simplicity giving only a few states to deal with. As a result, the previous researchers were able to immediately implement the value iteration or the policy iteration techniques and arrive at optimal policies. But in reality, if all the relevant issues of the medical problem were to be taken into consideration, the state space would grow quickly, requiring very high computation time.

1.7 Approach considered

1.7.1 Reinforcement learning (RL)

Instead of directly applying value-iteration or the policy-iteration algorithms, an indirect way to arrive at the optimal or, near optimal policy is by estimating a value function using the method of reinforcement learning on a simulation model of the problem. This is a viable alternative for obtaining near optimal policies for large scale MDPs with considerably less computational effort than what is required for DP algorithms. RL has two distinct advantages over DP. First, it avoids the need for computing the transition probability and the reward matrices. The reason being that it uses discrete event simulation as its modeling tool, which requires only the probability distributions of the process random variables (and not the one step transition probabilities). Secondly, RL methods can handle problems with very large state spaces since its computational burden is related only to the value function estimation, for which it can effectively use various function approximation methods such as, regression, and neural networks. Therefore, when the model of an environment can be simulated and

inputs such as rewards can be given, reinforcement-learning algorithm can be applied to get the optimal policy.

The hereditary spherocytosis problem that is considered in this thesis, has 1911 states and five actions. Therefore, the transition probability matrix is of the size (1911 × 1911). The idea is to simulate the model of the situation and embed it into the reinforcement learning technique. Thus, an optimal policy, which dictates, according to the patient's condition, what surgery to be performed and when it should be performed, can be obtained. Also, this process would give the physician an idea about the QALY (quality adjusted life years), the patient would enjoy, given, the optimal policy that is followed. Such a decision support system hopes to aid the physicians in the decision-making process.

1.8 Summary of remaining chapters

The rest of the thesis is organized as follows. Chapter 2 is the literature review, which discusses the existing literature on the medical decision problems, and the approaches, which the researchers took to model them. Chapter 3 discusses in detail about the problem being addressed and reveals the research objectives. Chapter 4 goes at length into the proposed methodology, assumptions involved and describes the proposed algorithm. It also discusses the future agenda. References and Appendices have been provided at the end.

CHAPTER 2

LITERATURE REVIEW

This chapter summarizes the existing research on the topic, "Medical-decision making for the class of intervention problems." It describes the work of selected researchers and the solution methodologies adopted by them. Thus, the chapter gives an idea, of the gradual progress in modeling and solving the decision problems from the domain of medicine.

Research on medical decision-making is about a decade and a half old and a fertile area for research. In this section, the different kinds of decision problems and techniques, which evolved to solve them, have been described. The problems related to medical interventions and clinical prognoses were being studied from long. But, formulating these problems as models, using statistical methods and artificial intelligence techniques began in the late 80's.

2.1 Decision trees

Early papers attempted to model the medical decision problems, in clinical settings, using decision trees. The reader is referred to Hollenberg (1984) [9] and Lau et al. (1983) [10] for further discussion on decision trees and recursive decision trees, respectively. But soon, it was realized that, this method involved assumptions, which were far from reality.

Sonnenberg and Robert Beck (1993) [11] explained, why decision trees and recursive decision trees, are not suitable to model decision problems in medicine. The following explanation is adopted from their paper. Decision problems involve an ongoing risk over time, because of which, there are two important consequences. One is the uncertainty of the times at which the events occur. Second, is the repetition of a given event. The decision tree modeling does not tell, as to when the events occur in time. Also, there is a problem of assigning utilities to the terminal nodes, because they do not represent an end but represent the prognosis of the patient for such an outcome, as is the node. The second consequence, that is the repetition of a given event, can be modeled by recursive decision trees. The problem in such modeling is that, the branches of the tree might increase exponentially with each repetition, making it impossible to track. Hence, Sonnenberg and Beck describe the markov model approach in this paper, which they felt was appropriate to model the decision problems. With the description of the use of Markov models for prognosis in medical applications by Beck and Pauker (1983) [12], Markov models have been applied and analyzed on many decision-making problems in medicine.

2.2 Markov cycle trees

In 1984, Hollenberg [9] introduced the Markov cycle trees, which have been used by some researchers in modeling. In 1993, Sonnenberg et al. [11], explained that Markov models are especially useful for decision problems, which involve risk, that is continuous over time. Methods were also described to evaluate markov models. It was concluded that, the ability of the markov models, to represent repetitive events and the time

dependence of both probabilities and utilities, allows for more accurate representation of the clinical settings. Three important ways of modeling in the Markovian manner were discussed. Namely, the matrix solution, the cohort simulation and the markov cycle trees. Also, the use of markov cycle trees was demonstrated by implementing the methodology to a case history of a 42-year old man, who had had a kidney transplant. While the patient was receiving normal immuno-suppressive drugs, a decision problem arouse. The continuation of drugs might give rise to a complication, but if the drugs were stopped, the kidney might be rejected. Therefore, the doctor had to decide on a treatment strategy, that maximizes the quality of life expectancy of the patient. The author by comparing, concludes that, Markov cycle trees are a suitable representation than decision trees. They also stated that, Markov cycle tree is a formalism that combines the modeling power of the Markov process with the clarity and convenience of a decision tree representation. The above-described medical problem was modeled by Kassier et al. (1988) [13] as a decision tree, prior to Sonnenberg and Beck.

2.3 Stochastic trees

At around the same time, Hazen (1992) [14] introduced, how medical decision problems based on age-dependant mortality rates and declining incidence rates may be modeled using stochastic trees. In this paper, it was shown that stochastic trees possess important advantages over the markov cycle trees for medical decision modeling. The stochastic tree is a continuous time version of a Markov cycle tree, useful for constructing and solving medical decision problems, in which risks of mortality and

morbidity may extend over time. Hazen (1993) [15] introduced the notion of factoring a large stochastic tree into simpler components, each of which may be easily displayed. This paper extends the idea of his previous paper, where stochastic trees were introduced.

2.4 Markov models

In the five part series of "Primer on medical decision making", authors, Krahn MD, Naglie G, Naimark D, Redelmeier DA, Detsky AS, (1997) [16,17,18,19,20] laid considerable emphasis on the Markovian way of modeling. Interested readers can refer to this excellent review, on decision problems and factors to be considered, while modeling them. Issues like, choosing an appropriate problem, determining the trade-off between accuracy and simplicity and deciding on a time frame have been discussed in Part 1 [16] of the series. Part 2 [17] of the series discusses, the construction of a decision theoretic approach for the giant cell arteritis (GCA) case. Part 3 [18] discusses the role of decision trees in modeling. Part 4 [19] describes how to derive probabilities and also describes bug proofing of decision trees. Part 5 [20] describes the same case as in Part 1, which has been modeled using Markov analysis. Though the authors suggest the Markovian way, they also leave a word of caution, that, model builders be aware of the pitfalls in this approach and suggest that the analyst must weigh the simplicity and clarity of a conventional tree against the fidelity of a Markov analysis. Part 5 concludes with the inference, that, there doesn't seem to be any significant qualitative difference between the markov approach and the simple tree approach.

2.5 Dynamic decision models

As different researchers tried to model the problems in different promising ways, Leong (1991) [21] attempted to model the medical decision problems, by focusing on the ontological features of the problem, like classes of actions, classes of events, classes of outcomes, probabilistic dependencies and temporal precedence. This attempt was made keeping in view, automating the construction of decision models in medicine. The proposed system, described in this paper consists of a planner, which constructs a decision model by accessing the medical knowledge database, and solves the model. The solution is given to the user. The user helps the planner in doing its job, by giving certain inputs. The results of this paper show that, to support dynamic decision modeling, the structure of the knowledge base, must reflect the nature of both the decision problem and the domain knowledge. Qualitative probabilistic network was used in modeling.

Leong (1992) [22] tried to represent knowledge, which is based on the context of the problem, as a network. She believed that, complexity in the medical problem's knowledge occurs due to the variations in the contexts of the underlying phenomenon. In that way, a framework has been proposed, which attempts to model the uncertain knowledge in network formalism. In this paper, she explains, how to represent uncertain situations in a network form, various components of the network and its applications. She worked with the different structural relations, uncertain or behavioral relations, context dependant notions and different relevant phenomenon of the problem, to model it as a decision problem, though the implementation was left to be done.

Leong (1993) [23] identified that Semi-markov decision process (SMDP) can be taken as the common theoretical basis for solving the decision problems. Until this point of time, simple Markov decision process (MDP) has been in use. In this paper, it was explained that the complexity involved in the decision modeling could be avoided by dealing directly with its underlying mathematical framework like an SMDP, which would be more near to the practical situation. In an SMDP, the duration for which a patient is in a particular state, which is yet another dimension of uncertainty, can also be taken into consideration. It was also pointed out that, though, there are different formalisms suitable for different kinds of problems, it should be realized that the underlying mathematical framework for solving any of these problems is the same. It is either an MDP or an SMDP. In this paper, the example of a typical medical decision problem, "The management of chronic ischemic heart disease" was considered and modeled using three different formalisms, namely, dynamic influence diagrams, stochastic trees and Markov cycle trees. The pros and cons of the formalisms, were discussed and the paper concludes with the notion, that, difficulty in modeling medical problems is not with the formalism, but, is with the computational complexity of the value-iteration or the policy-iteration technique of the underlying dynamic programming formulation, which cannot be avoided. This paper can be considered as an important milestone in the research related to this area.

In an attempt to provide a general framework for modeling and solving decision problems, Leong (1994) [24], came up with a framework called, "Dynamic decision modeling language" (DynaMol). The idea behind this, as she explains, is to have a

general framework, which can handle any type of graphical formalism, as long as the underlying methodology is an SMDP. According to the paper, the framework provides a unifying task definition and a common vocabulary for the relevant decision problems and also balances the trade-off between model transparency and solution efficiency in the current frameworks. In this paper, Leong essentially describes the DynaMol design, the dynamic decision grammar, which, comprises of terms related to modeling, the graphical representation convention and the solution methods. The paper also summarizes the assumptions involved in the design of DynaMol such as,

- Same states should be valid through out the decision horizon,
- Same set of actions is applicable in each state,
- Transition probabilities can vary with time,
- Semi-Markov decision process has limited memory regarding the past events. But in some cases, the memory about previous states and actions could be important. Leong notes that DynaMol should be extended to take care of such things.

Cao et al. (1996) [25] discusses, issues like the requirement for a multiple perspective dynamic decision modeling language, the design of DynaMol framework, the semantics and the grammar. Further literature on the same topic, can be obtained, in the technical report by Leong (1994) [26]. This contains all the work done by her, in the area of medical decision making until the year 1994.

Leong (1996) [27] explained DynaMol in detail and implemented it on the "Atrial fibrillation" case. The problem was modeled using influence views and SMDP as the underlying framework. DynaMol models the problem, translates into the grammar of the underlying framework, solves and finally analyses it.

Leong (1996) [28] illustrates, further improvements in the DynaMol design, which accommodates "translators". Graphical representations often help the analyst in understanding and in easily accommodating all the complex factors of the medical problems. But there are various types of graphical formalisms, like the influence views, transition views, and markov cycle trees. Depending on the analyst, the problem can be represented using any of the above and can be fed to DynaMol. DynaMol, then, translates that particular graphical formalism, first to a transition view and then to the underlying mathematical framework. This translation convention has been elaborately discussed in this paper and the present DynaMol design was implemented and tested on the case study of the atrial fibrillation case.

In 1998, Cungen Cao et al. [29] proposed a technique, through which diagnostic test strategies can be obtained. This technique is very different from the MDP modeling and uses the artificial intelligence techniques. It is similar to the decision tree technique and gives a diagnostic test strategy from medical data. The authors call this modeling, a 'strategy tree'. This tree can be induced from three types of information measures, namely, K-level information gain, K-level gain ratio and K-level cost effectiveness. The test, which provides the most information, has a larger information gain ratio and thus, selected. The induction of the strategy tree depends on the previous tests selected. The

cost of the test strategy is taken into consideration, to resolve in case of two tests of same information gain ratio. Cost, here, is the reward obtained. In the authors' words, the building of the strategy tree is more or less similar to a decision tree building, except for the difference, that the tree is also built in a level-by-level manner, in addition to the divide-and-conquer manner.

Sunderesh et al. (1999) [33] extended the DynaMol framework, by embedding abstraction mechanisms, which allows the end user to switch between representations of the medical problem. This is called abstract modeling, which gives guidance to the user, through the involved constraints in the problem.

Harmanec et al. (1999) [34] attempted to model the problem of "Severe head injury management", using a simple influence diagram. The decision problem involved was to prescribe an optimal treatment plan to a severe head injury patient in an ICU setting. This problem is different from other decision problems, considering its criticality and large number of complex factors and parameters varying in minutes. Two ways of parameter elicitation were proposed and the authors concluded that, more efficient strategies for obtaining the numerical parameters involved are needed, even though the problem produced reasonable recommendations.

An excellent critical review paper, came into the research area of medical decision problems, when, Peter Lucas et al. (1999) [35] described, the various decision-making methodologies, used in the field of statistics and probability and in artificial intelligence (AI). In this paper, restricted probability models, decision trees and Markov

processes have been grouped under the statistical methods. Neural networks and Bayesian belief nets were grouped under the AI techniques.

Qi and Leong (2001) [36] set up a method, for automatically constructing influence views for the medical problems, directly from data. The conditional probabilities for the influence views can also be automatically generated, using Bayesian approach described by Cao et al. (1997) [37]. This methodology was accommodated in DynaMol.

In the two papers, Lin et al. (2002) [38] [39] solved two problems, namely, "Spontaneous pneumothorax" problem and the "chronic cough" problem, using the SMDP modeling, which she proposed earlier and represented the problems in the influence view formalism. Also, in 2002, YP Xiang and KL Poh [40] published a paper, which models medical problems, which are time critical in nature. Usually, for decision analysis, it takes considerable time. But in critical medical problems, the decision has to be taken in a matter of minutes and that adds, the constraint of limited time, to the decision problem. To formulate such problems, Xiang and Poh, proposed, a time critical dynamic influence diagram (TDID), which can represent both space and time abstractions within the model. Further, they proposed four algorithms to solve the TDID's. The authors follow a meta-reasoning approach to select the appropriate algorithm, from the four algorithms, in terms of computational complexity and decision quality. This methodology was implemented on a cardiac arrest problem and the results looked promising.

2.6 Obtaining the numbers

In the 1990's, while various methodologies were being proposed for modeling the medical decision problems, research for obtaining the required numerals (probabilities) used in the models as inputs, was also progressing. The extraction of transition probabilities and the action rewards, required in modeling, became an important topic of research. The transition probabilities needed for the MDP, has to be, either obtained from the domain experts or have to be extracted from the medical databases.

Cao and Leong (1996) [25] attempted to automate the learning of transition probabilities and action rewards, required in the modeling of an MDP, from the medical databases. It was suggested in the paper, that static comparison is an efficient method to extract the transition probabilities, in which the transition cases have been divided into three semantic classes. Using this method, the paper claims, that the issues of incomplete and infrequent databases can be overcome to a considerable degree.

Cao et al. (1997) [37] proposed a Bayesian method, for automated learning of conditional probabilities, from large medical databases. Obtaining probabilities from domain experts, also, has been analyzed. Several issues on pre-processing raw data, for applying to the decision problems were discussed. The learning from databases of probabilities is based on the DynaMol framework. The proposed methodology was implemented to the problem of colorectal cancer and results have been obtained.

In 1998, Cungen Cao et al. [41] published his Bayesian approach, to automatic generation of conditional probabilities and its results. Lau and Leong (1999) [42], proposed a framework, which can obtain the probability distributions for the decision

problems from domain experts. These distributions are very important, as they represent, the uncertainties in the system. This framework involves the doctors in getting probabilities and also tries to minimize the bias in the probabilities given by them.

Zhao (2000) [43] proposed, an automated data pre-processing framework, which uses database scripts, for processing databases before eliciting probabilities for dynamic decision models. Thus, the eliciting of the numbers is by itself, an interesting area of research in the domain of medical decision-making.

2.7 Static modeling

DT-Planner is a software package written in Ansi-C language. This is developed by Paolo Magni et al. (1997) [6] to design and solve dynamic decision problems. It makes use of influence views, to represent the problem. A user-friendly graphical user interface, allows the user to navigate through the built in menus, to draw the influence view of the problem and to input the conditional dependencies, involved, between the events of the problem. The software models the problems as an MDP and then calculates the transition probability matrix. DT-Planner solves the problem, using the value-iteration algorithm to find the optimal policy. Elimination algorithm, by Rina Dechter (1996) [44], is used to remove event variables from the influence view and to compute the equivalent MDP. The problem of "allogenic bone marrow transplantation" has been implemented, using this software and the optimal policy obtained was convincing.

The problem of the "Heriditary Spherocytocis" (HS) disease has been lurking through the minds of researchers for quite some time. Patients with mild HS, have an increased risk of gall stone formation and complications. Various treatments are

available, out of which, Marchetti et al. (1998) [45] considered, three treatment strategies, namely, ‘splenectomy’, ‘cholycystectomy’ and ‘no surgery’, so that the problem can be simplified. A decision analysis was performed to see the effect of the three strategies, on the quality-adjusted life expectancy.

The problem was modeled in the form of two phases. The first phase was modeled as a decision tree, beginning with a decision. The outcomes of that decision, depicted, surgery related mortality and accommodated, compliance to and adverse effects of prophylaxis against infection. The second phase was modeled as a Markov cohort analysis. But this didn’t serve the purpose of modeling the problem anywhere close to the reality, as the model represented a static situation in the first phase and hence, the dynamic element of the problem has been discarded.

Static modeling, requires the decision model to solve the problem at any age, as if it were the only possible decision time, without considering the other decision time points and hence, that the decisions might be reconsidered later. Also, the model proposed by Marchetti et al. (1998) [45], allows, only one chance to take a decision and that too, immediately.

Paolo Magni (2000) [8] modeled the above problem, by removing the two phases and as an MDP, using influence views. The influence views and conditional probabilities were fed to the DT-Planner (described above), to be solved by value-iteration technique and arrive at an optimal policy, which maximizes the quality-adjusted life expectancy of the patients. The results obtained, showed little improvement, when compared to the static model and hence, an issue of investigation.

Paolo Magni considerably simplified the HS problem, by making many assumptions, many of which were far from reality. Also, the MDP model doesn't seem, either appropriate or accurate. Also, the calculation and consideration of the utility values, which are in quality adjusted life years (QALY) is not very clear and convincing. As such, medical problems are complex and the case of HS, is one of them. It seems to us that proper modeling of this problem, as an MDP would result in a large state space, to the order of $9.12 * 10^5$. But, the model by Paolo et al. has a total of 11 states. As the state space became dwarfed, the age-old dynamic programming algorithm (value-iteration technique) could be applied and solved for an optimal policy using the DT-Planner. However, for a large state space problem (as mentioned earlier in the introduction), value-iteration technique would take forever to solve and would barely help.

Moreover, until now, in the literature, researchers have been modeling any kind of decision problem as an MDP and solving it with only the available value iteration technique.

This situation challenges us and motivates to propose a methodology, which can model and solve any kind of medical decision problem, especially the ones with large state spaces. We choose the ever-interesting HS problem for our research, summarized in the following chapters.

CHAPTER 3

RESEARCH OBJECTIVES

In this chapter, the problem of Hereditary Spherocytosis and its symptoms are described, and the research objectives are stated.

3.1 Problem statement

The problem under consideration is the Hereditary Spherocytosis (HS), which is the most common erythrocyte membrane disorder. Patients with this disease, suffer from a chronic destruction of red blood cells. It is known, that in 60% of the cases, the disease is severe and the patients become extremely anemic. In the rest 30% of the cases, the patients are mildly anemic, with a hemoglobin level over 11 g/dl, a reticulocyte count of 3-6% and a bilrubin level of 1-2 mg/dl. These patients have an increased risk of gallstone formation, because of the sustained erythropoesis, which predisposes them to episodes of parvovirus induced aplasia and haemolytic crisis.

In the severely anemic patients, performing surgery, called splenectomy and removing the site of red blood cell destruction is mandatory. But, in the case of mildly anemic patients, there is no necessity to perform splenectomy immediately. These patients have other treatment options available other than splenectomy. Thus, arises a decision problem for the physician. Keeping in view, the side effects of splenectomy and the availability of other treatments, the physician gets into a dilemma as to which would be the best decision. He/she has to trade-off between, preventing adverse disease consequences, and the risks posed by surgery, including, mortality, morbidity and post

splenectomy infections. The other available treatments would comprise of no surgery, where the patient is not treated but kept under observation to intervene at a later point of time and the laproscopic cholycystectomy, which can prevent gallstone formation.

Therefore, the decision problem consists of coming up with the optimal therapeutic plan, which dictates, depending on the patient's condition, what surgery should be performed, when it should be performed and in what condition can it be performed, to maximize the quality of life of the patient under consideration.

3.2 Research objectives

The objectives of this research are the following,

- to propose a methodology, which accommodates the modeling of the sequential decision problems in medicine, as a MDP, and, to use a computer simulation based reinforcement learning algorithm for an efficient solution,
- to model the HS disease problem as a MDP and to obtain the results by solving it using the proposed algorithm,
- to compare the results obtained by the proposed methodology algorithm, with the results obtained using dynamic programming algorithm.

CHAPTER 4

PROBLEM FORMULATION AND SOLUTION METHODOLOGY

The Hereditary Spherocytosis problem has been formulated as a MDP. This chapter describes in detail, the issues of modeling the problem as a MDP and its solution methodology using a reinforcement learning algorithm. The simulation mechanism involved and the 'average reward reinforcement learning' algorithm are also presented.

4.1 Problem formulation

Let the system state of a patient be described by the vector '*s*'. The system state consists of the basic variables necessary to describe the patient's state. These can be called as the state variables and in every decision epoch, the physician chooses an optimal action based on the current state of the patient. The important elements of the patients' state are the following variables.

- Presence of gallstones
- Presence of spleen
- Presence of sepsis
- Time elapsed after splenectomy is done (in years)
- Presence of complication
- Age and sex of the patient

Therefore, the system state vector can be written as

$$s = (g, s, \tilde{s}, t, c, a, \hat{s}), \tag{4.1}$$

where,

- g, describes the state of gallstones,
- s, describes the presence or absence of spleen,
- $\tilde{s}$, describes the presence or absence of sepsis,
- t, describes the elapsed time after the surgery splenectomy is performed,
- c, describes the presence or absence of a complication,
- a, describes the current age of the patient at that particular decision epoch,
- $\hat{s}$, describes the sex of the patient.

The underlying Markov chain of the MDP can be denoted by

$$S = \{S_n : n \in N, S_n \in \xi\}, \tag{4.2}$$

where,

- S_n, denotes the system state at the n^{th} decision epoch,
- n, of the decision epoch index,
- ξ, denotes the state space,
- $N \in \{1,2,3,....,100\}$.

(See Appendix A for a detailed description of a MDP).

At any decision epoch n, $S_n \in \xi$ and the action is chosen as $a_n \in A(s)$,

where, $A(s)$ denotes the set of all possible actions in a state 's'.

4.1.1 Elements of the MDP

The five important elements of an MDP are defined below for the problem under consideration.

4.1.1.1 State space

The values associated with the state variables are as follows,

- g = {1, 2, 3, 4, 5, 6},
- s = {0, 1},
- $\tilde{s}$ = {0, 1},
- t = {0,1, 2, ….95},
- c = {0, 1, 2},
- a = {1, 2, ….100},
- $\tilde{s}$ = {0, 1}.

Therefore, the cardinality of the system state space ξ is

$|\xi| = 6 * 2 * 2 * 95 * 3 * 100 * 2 \approx 13.68 \times 10^5$.

Total number of states in the associated transition probability matrix = $187.1424 * 10^{10}$.

4.1.1.2 Action space

The action vector is given as $A = (a_1, a_2, a_3, a_4, a_5)$ where, a_i, $i \in \{1, 2, \ldots 5\}$ denotes the five intervention strategies. Every year, the physician can choose among the following strategies.

- a_1, no prophylactic surgery
- a_2, prophylactic splenectomy
- a_3, prophylactic cholycystectomy

- a_4, prophylactic splenectomy and prophylactic cholycystectomy
- a_5, open surgery (in the case of a complication occurring due to gallstones)

Not all of these five action choices, though, are available for every state. Therefore, the availability of these actions depends on the state in which the patient is present.

4.1.1.3 Time horizon

In the present model considered, the maximum life span of a patient is assumed to be 100 years. However, from every state $s \in \xi$, there is a nonzero probability of natural death for the patient, apart from the probability of treatment related death associated to certain states. Therefore, a patient is assumed to live for 100 years or less.

4.1.1.4 Decision epoch

The time between two decision epochs is considered to be 1 year. It is assumed that a patient visits the doctor every year and the patient state is observed every year. Therefore, the normal life span of a patient in years would equal the number of finite decision epochs the Markov chain evolves through before reaching the state of death. However, the quality adjusted life years that the patient enjoys is calculated using the utility function and is different from the normal life span.

4.1.1.5 Transition probabilities

For every action $a_i \in A$, there is a transition probability matrix $P(a_i)$ of the Markov chain S, where $P_{ss'}(a_i)$ represents the probability of moving from state s to s' under action a_i. These transition probabilities can be obtained from domain experts or abstracted from medical databases. Transition probabilities are assumed to be stationary.

$$P_{ss'}(a_i) = P\left\{\frac{s_{n+1} = s'}{s_n = s, a_n = a_i}\right\}. \quad (4.3)$$

4.1.1.6 Rewards

To obtain the best strategy, there has to be some measure of an action's value, so that one can compare different actions. Hence, an immediate value is specified for performing each action in each state.

Given the system state s at decision epoch n and action a_i, if the next state is s', the expected value of the reward is

$$R_{ss'}(a_i) = \left\{\frac{\gamma_{n+1}}{s_n = s, a_n = a_i, s_{n+1} = s'}\right\}. \quad (4.4)$$

The rewards can be in any unit of interest. For example, monitory cost, lifespan or cost-effectiveness ratios etc. The rewards are considered here as the Quality Adjusted life Years (QALY) of the patient.

4.1.2 Quality adjusted life years (QALY)

The following sections describe quality of life adjustment, define a QALY and explain various methods to derive quality weights for health states. Then, the proposed method to obtain quality weights for health states is discussed. Finally, the procedure of obtaining immediate rewards in terms of QALYs has been explained.

4.1.2.1 Utility function

The utility function used to compare the different strategies is based on the Quality Adjusted Life Years (QALY) concept. Quality of life adjustment, measures the degree to which surgical interventions, medical therapies and disease states diminish the well being of patients. It is expressed as a number between 0 and 1, for every health state.

The physician's decision, coupled with the inherent changes occurring in the health of a patient, can lead the patient to a decrease or increase in his/her quality adjusted life years. A patient's QALYs are observed every year, until the patient dies and the overall gain or loss in QALYs is calculated. Thus, the objective function considered, is to maximize the gain of the QALYs over the patient's whole life. Hence, the utility function considered in this thesis is based on the QALY concept. The assumption involved associating the value function, concerning the utility is that, it is time separable. This implies, that it would be possible, to calculate the overall value or the utility function as a combination of functions, specified at each decision epoch.

4.1.2.2 QALY

According to Joshua graff Zivin (2002) [52], economists prefer to measure any physical quantity in terms of their monitory value. But, health economists didn't prefer such a method because of the general belief that life is too precious to be priced, or that, such a pricing is morally unacceptable. Therefore, health economists relied on other methods, which measure the benefits from any health related activity that affects health, in units of health outcomes. These units could be blood pressure units, cases of a particular disease or life years. That's how, quality-adjusted life years emerged as a measure for health related outcomes.

Anytime when one talks about the outcome of a treatment or anything which effects the health of a person, there are always two issues involved. One is the 'quantity' of life of the person affected because of the intervention and the other is the 'quality' of life of the person, which measures not the number of years a person lives, but measures

the years, which he/she lived comfortably with perfect health. The problem with using only quantity of life as a measure is that it only considers whether people are alive or not and is often expressed as life expectancy. On the other hand, quality of life takes care of a number of issues concerning people related to their physical and mental capacity coupled with the emotional aspects. Formally, a measure of quality of life is a quality adjusted life year. In mathematical terms, it can be expressed as,

QALY = Life Expectancy x Quality of the remaining life years.

The quality of the remaining life years is quantified by placing a weight on time in different health states. The concept of QALY provides a common basis to compare the different kinds of interventions in terms of health related quality of life.

The idea of QALY is explained in brief below with the help of an example. Suppose, a physician is trying to decide between two treatment strategies. Treatment A has more probability of treating the disease than treatment B, but A leads to side effects whereas B, has no significant side effects. Then, to compare the benefits of the two treatments, the physician needs to know more than just the probability of success of each treatment or life years saved. He/she should also know the amount of value which people place on the health state with side effects related to the treatments, the quality weight for the health state with side effects. This quality weight is also sometimes referred to as the utility value of that particular health state. Suppose, the quality weight for the state with side effects was estimated to be 0.75 (i.e.,) one year with side effects is equivalent to 9 months in perfect health. And if treatment A yields a total of 10 extra life years, then it is

said that treatment A yields $0.75 \times 10 = 7.5$ QALYs. This figure would then be compared to the QALYs generated from treatment B to determine which yields greater benefits.

4.1.2.3 Methods for deriving quality weights for health states

Research on the subject provides different methods to generate the quality of life values or the quality weights, by observing the health states of the patients. These are also often referred to as preference weights or utility values of the states. The methods to obtain these weights fall into two broad categories. First category comprises of ,

- rating scale technique,
- standard gamble technique,
- time trade off method.

Second category comprises of the multi-attribute health states survey.

The methods in the first category directly assess the quality weights with the help of preferences of the individuals for well-described health states. Here, individuals are asked to rank the given health state, relative to death and perfect health. The way of ranking the health states differs from one method to the other. The second category methods break down all the health states into a set of primary quality attributes, which characterize those health states. Individuals are then asked to fill questionnaires designed specially for the purpose. These multi-attribute survey answers are transformed into quality weights using the first category techniques.

Below given is a brief description of the methods in the first category as explained by Joshua graff Zivin.

4.1.2.4 Rating scale

In this method, individuals are provided with a set of health states and are asked to select the best and worst of those states. Then, those two states form the minimum and maximum rating on a scale, usually the best being 1, and the worst state taking the value zero. All the other states are placed on the scale according to the rating of the individual. After this the respective ratings of the states are converted into quality weights of the respective health states.

If, death is the worst state,

$$\underset{ForAnyGivenHealthState}{QualityWeight} = \frac{ScaleValueOfTheHealthState}{ScaleValueOfTheBestPossibleState} \quad . \tag{4.5}$$

If, death is not the worst state,

$$\underset{ForAnyGivenHealthState}{QualityWeight} = \frac{x - y}{z - y} \quad , \tag{4.6}$$

where, x = Scale value of the health state,

z = Scale value of the best state,

y = Scale value of death.

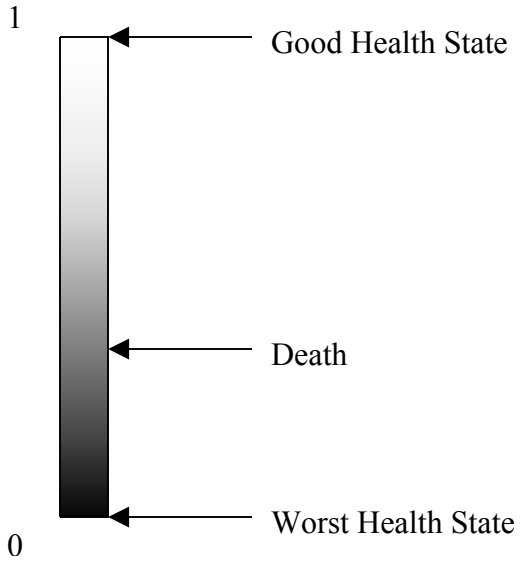

Figure 5. Rating scale for quality weights

4.1.2.5 Standard gamble

In this method, the subject is asked to choose between two alternatives. The first alternative has two possible outcomes.

- Perfect health state of quality weight 1, for a length of time '*t*'
- Immediately going to worst state of quality weight 0

The second alternative is living in the same imperfect health state for a time '*t*' with certainty. The probability of perfect health is denoted by '*p*' and the probability of going immediately to worst state is denoted by '1 – *p* '. '*k*' denotes the probability of being in imperfect state. The quality weight for state 'imperfect health' is determined by varying the probability '*p*' until the subject is indifferent between the two alternatives. The weight for state '*k*' is equal to the value '*p*'.

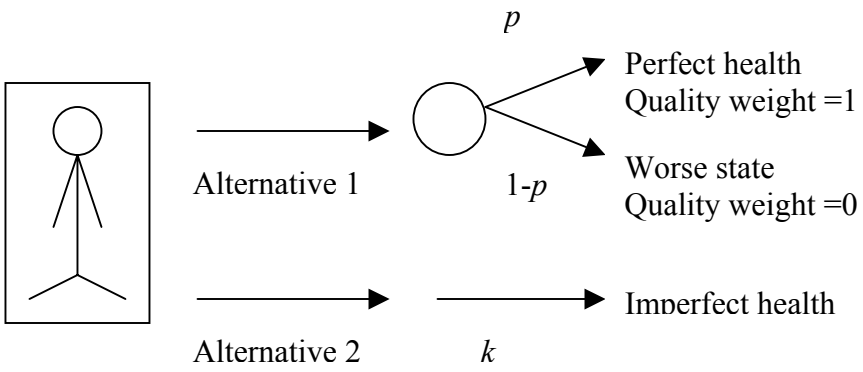

Figure 6. Standard gamble for deriving quality weights

This technique has got the uncertainty element in it. The quality weights are determined by the risk of going to a worse state, that an individual is willing to accept to get an improvement in his/her health.

4.1.2.6 Time tradeoff

This also, has two alternatives for the subject to choose from. Alternative 1 is life in imperfect health state '*k*' for time '*t*' and then death. Alternative 2 is perfect health for time '*y*' and then death. But, $t > y$. time '*y*' is varied until the subject is indifferent between the two alternatives.

$$QualityWeightForState'k' = \frac{y}{t}. \tag{4.7}$$

The aim of this method is to determine the amount of life expectancy an individual is willing to sacrifice to increase the quality of their health.

4.1.2.7 Multi-attribute health status surveys

In this method, the health states are characterized by important health attributes. For example, the EuroQol system contains five attributes, namely, mobility, self-care, anxiety/depression, pain or discomfort and usual activities. These attributes further have levels, from which the subject chooses, according to the health situation. Under mobility, for example, the subject can choose from,

- no problems walking,
- some problem waking,
- confined to bed.

Then, these levels, which the subject chooses according to the health condition, are transformed into quality weights in two steps. First step involves, determining a method for aggregating the attributes and specification of a multi-attribute utility function. Second, a large number of people are given questionnaires, which are designed to incorporate all the attributes and the people are supposed to check the attributes with which the health state can be defined. The same population is also asked to weigh the health state using the standard gamble or the time tradeoff methods. Then, the two sets of quality weights of health states are used to estimate the parameters of the multi-attribute utility function. In the end, the result is a set of quality weights for all the possible attributes and levels in the questionnaire, allowing any pattern of answers to be assigned into a single quality weight that is bounded between 0 and 1.

4.1.2.8 Cost-utility ratios

When the QALY values are combined with the costs of the interventions, cost-utility ratios can be obtained which can be used as another measure for differentiating between interventions.

Mathematically,

$$CostUtiltyRatio = \frac{CostOfInterventionA - CostOfInterventionB}{No.OfQALYsByInterventionA - No.OfQALYsByInterventionB}. \quad (4.8)$$

These ratios indicate the additional costs required to generate a year of perfect health (i.e.,) one QALY, through an intervention.

Though, all these methods are available to determine quality weights to the health states, they all involve, a population of subjects. In the present thesis, because of lack of access to actual individuals suffering from HS and also due to lack of resources for conducting statistical surveys over a population of subjects, the above mentioned methods cannot be followed to obtain quality weights. However, a method has been developed for the purpose, which produces quality weights that are most consistent with the health states involved in the model. This method is more on the lines of the multi-attribute survey technique. This has been developed to obtain the quality weights, so that, the proposed methodology can be checked for validity. Nevertheless, the methodology is capable of incorporating quality weights obtained by any method available.

Research in area of 'evidence based medicine' is exploring ways to come up with consistent mathematical methods to measure quality of life. For example, Bernard M. S. van Praag and Ada Ferrer-I-Carbonell (2001) [53], discuss how QALY losses can be assigned to various impediments and illenesses. A mathematical methos has been

proposed to calculate the QALYs based on the age of the person and the results of the paper shoe that the method is operationsl to evaluate the health situations of populations and population subgroups. Nevertheless, the use of QALYs in decision-making does mean that the different kinds of interventions are being distinguished from each other and the differences between them made explicit.

4.1.2.9 Limitations of QALYs

QALYs are a mere indication of the benefits of a particular intervention. These values could be far from being perfect as a measure of outcome. The following are some of the limitations of QALYs,

- lack of sensitivity when comparing two similar drugs, which are competitive,
- preventive measures where the impact on health outcomes may not occur for many years may be difficult to quantify using QALYs,
- QALYs are highly dependant on age and life responsibilities. For example, it is difficult to compare an athlete's ankle fracture with that of a young boy, who have been restored to some degree of mobility,
- definition of perfect health is highly subjective.

Nevertheless, this procedure can aid anyone, wanting to use the system, at least in prioritizing their expenditure, while choosing from a variety of interventions. New techniques and therapies are bringing in much complexity for the health care professionals as to which strategy to choose. Therefore, the concept of QALY and cost utility ratios provide additional information, thus aiding the health care professionals in decision-making.

4.1.2.10 Uses of QALYs

The concept of QALY is used more as a comparison tool.

- It can be used to identify public health trends for therapies to be developed
- To assess the effectiveness of health care interventions
- To determine state of health in communities

4.1.2.11 Method followed to derive quality weights for health states

As mentioned earlier, the method followed in the present thesis, to derive quality weights to health states is similar to the multi-attribute method described. The health state of the patients with HS can be characterized by the five attributes, namely, gallstones (g), spleen (s), sepsis ($\tilde{s}$), time (t) and complic (c). Further, these attributes have their respective levels. Because of lack of resources and time, a statistical survey has not taken place with the help of questionnaires. Nor, was there any sort of input from general population regarding quality weights using standard gamble or the time tradeoff methods. Hence, the parameters of the multi-attribute utility function are not estimated by comparing the answers from the general populace, but are assigned some arbitrary values. These values, though arbitrary, are consistent in deriving a set of quality weights for each possible level chosen, according to the state of the patient, thus allowing any pattern of answers to be aggregated into a single quality weight that is bounded between 0 and 1. The consistency involved in obtaining the weights for all the health states involved in this model, makes it promising to use in the present modeling methodology and to check its validity.

This method is further described in detail. The different attributes and their respective levels, along with the arbitrary values, which they yield, have been shown in Table 1. Depending on the health state, values are attained for all the five attributes, according to their respective levels. These values are then summed up to arrive at the quality weight of that particular health state. In this manner, the quality weights for all the states (2685) have been derived.

Example 1

The quality weight for the health state *s* (3,1,0,0,1) would be (from Table 1)

0.15 + 0.0 + 0.10 + 0.05 + 0.0 = 0.30.

Example 2

The quality weight for the health state *s* (2,0,1,5,0) would be

0.20 + 0.30 + 0.00 + 0.0026 + 0.30 = 0.803.

Table 1. Quality weights

Note: All values are in generic units

Attribute	Level Description	Level	Value
Gallstones	No Gallstones	1	0.22
	Asymptotic Gallstones	2	0.20
	Occasional colics	3	0.10
	Recurrent Colics	4	0.07
	Gallbladder removed	5	0.15
	No Gallstones (Death)	6	0

Table 1. (Continued)

Attribute	Level Description	Level	Value
Spleen	Present	1	0
	Absent	0	0.22
Sepsis	Present	1	0
	Absent	0	0.23
Time	Splenectomy not done	0	0
Time (if sepsis = 0)	Splenectomy done	1 year	2 * 0.15625
	Splenectomy done	2 years	3 * 0.15625
	Splenectomy done	: years	: * 0.15625
	Splenectomy done	: years	: * 0.15625
	Splenectomy done	95 years	96 * 0.15625
Time (if spleen = 1)	Splenectomy done	1 year	15
	Splenectomy done	2 years	15 – (1 * 0.15625)
	Splenectomy done	3 years	15 – (2 * 0.15625)
	Splenectomy done	: years	15 – (: * 0.15625)
	Splenectomy done	: years	15 – (: * 0.15625)
	Splenectomy done	: years	15 – (94 * 0.15625)
Complic	Present (due to gallstones)	1	0
	Present (due to spleen)	2	0
	Absent	0	0.18

4.12.12 Immediate rewards in terms of QALYs

The immediate rewards obtained are in terms of QALYs, when a state transition occurs. Suppose, if a patient is in state 1, with quality weight 0.3 and an intervention takes place. Due to the effect of the intervention, coupled with the body's natural metabolic rate, if he is found to be in state 2, with quality weight 0.8, then it is believed, that the patient led the one year period within a health state of quality weight of 0.8 – 0.3 = 0.5. If the patient were to continue in state 1 for the one year period, without transiting to state 2, because of the intervention, then the QALYs he would have enjoyed, would be 0.3×1 year = 0.3 QALYs. Another perspective can be the one of the patient to be in state 2, right from the beginning, for the one-year period. Then, his QALY would have been 0.8×1 year = 0.8 QALYs. But, in the case of the patient's transition from state 1 to state 2, he gained some quality weight owing to the transition, which occurred due to the intervention. Thus, he gained 0.5 QALYs by transiting from a state, which provides 0.3 QALY's to a state, which provides 0.8 QALYs. This gain in the QALYs is considered as the immediate reward, due to the respective intervention.

These immediate rewards are aggregated to get the total expected reward, at the end of the Markov cycle, which here, is indicated by the death of the patient. Hence, the objective function, here, is to maximize the gain in the QALYs of a patient. These, total expected rewards obtained for each patient is compared and the patient whose total

expected reward is the highest, is selected. The policy, followed by that patient, becomes the optimal policy, which dictates what action to be taken in what states, such that the gain in the QALYs is maximized.

4.1.3 Hereditory spherocytosis

In this section, more details of the Hereditory Spherocytosis along with some assumptions are given. This information is needed for simulating the treatment process.

4.1.3.1 Spleen

Spleen is the red-blood cell destruction site (detailed in the problem description), which can be removed with the help of surgery. The presence of spleen poses an increased probability of gallstone formation. The absence of spleen causes a high risk of infectious condition known as sepsis. The incidence of sepsis depends on the length of time, since the spleen has been removed by the surgery splenectomy. Less risk has been associated for sepsis formation, of less than or equal to 4 years of spleen removal. More risk is associated for the formation of sepsis, after 4 years of spleen removal. Spleen can be removed with the help of surgery. At each decision epoch, a patient without spleen will remain in the same situation, that is without spleen and a patient with spleen also shall remain in the same condition, unless an action is taken to intervene the condition. It is assumed that a patient visiting the doctor for the first time, would not have undertaken any kind of prior treatment, or would not have undergone any surgical procedure relating to his/her disease.

4.1.3.2 Gallstones

The gallstone history of a typical HS patient can be classified as follows. The gallstones state variable has been assigned levels depending on the state of gallstones of the patient. The corresponding levels are shown in parenthesis below.

- Patients without gallstones (level 1)
- Patients with asymptotic gallstones, i.e. gallstones found through ecography but without clinical procedures (level 2)
- Patients with gallstones and occasional biliary colics, i.e. less than three episodes in the last year (level 3)
- Patients with gallstones and recurrent biliary colics, i.e. more than three episodes in the last year (level 4)
- Patients without gallbladder, because it has been removed (level 5)
- Patients who are dead (level 6)

Hence, the gallstone state variable takes the values from 1-6. After each decision epoch of the Markov chain,

- a patient can remain in the same state of gallstones,
- can develop asymptotic gallstones,
- can develop occasional biliary colics or recurrent biliary colics, if he/she already has asymptotic gallstones, or
- can develop recurrent colics if he/she has already occasional colics.

Gallstones cannot develop if the gallbladder has been removed. A transition diagram of the gallstones is shown in Figure 7. Gallstones can be removed with the help of surgery. It is assumed that the presence of spleen increases the risk of gallstone formation.

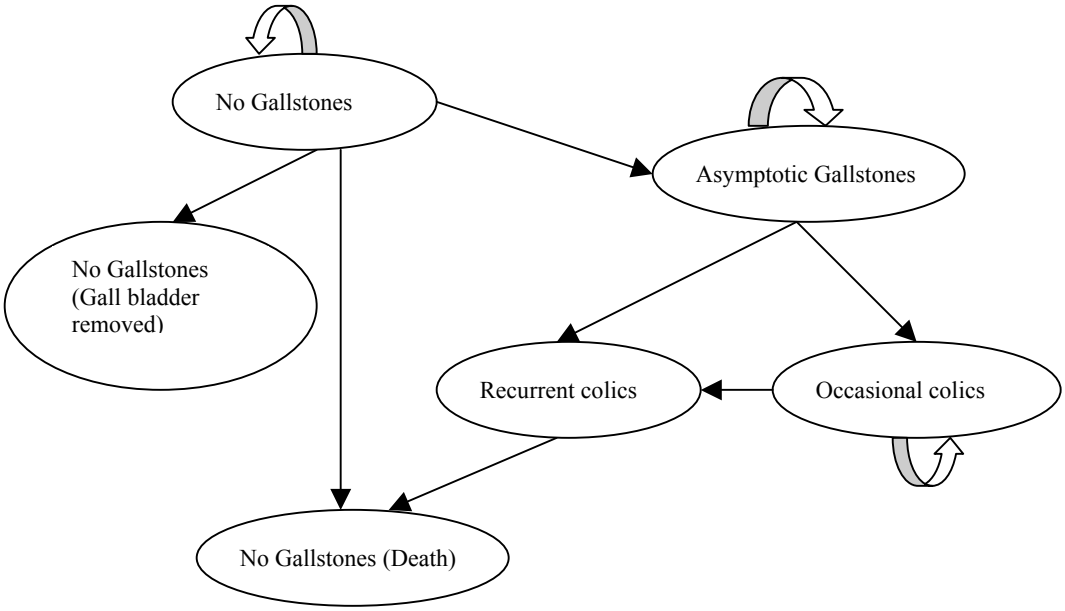

Figure 7. State transition diagram for gallstones

4.1.3.3 Sepsis

The patients at any point of time can be classified on the basis of sepsis as,

- patients who developed sepsis (level 1) and,
- patients without sepsis (level 0).

As mentioned previously, the condition of sepsis occurs only when spleen is absent in a patient and the risk of sepsis formation is dependant on the time elapsed after the spleen removal by splenectomy. It is assumed that surgery cannot be done if a patient develops sepsis.

4.1.3.4 Time

The time that elapses after the surgery splenectomy is kept in track. As said before, there is less risk of formation of sepsis within 4 years of doing splenectomy and more risk after 4 years. Therefore, in the simulation of the proposed framework, this issue has been taken into account to give an idea as to how much time (in years) has elapsed since splenectomy was done and accordingly, the risk of sepsis in the form of probabilities has been assigned to the transitions occurring from one patient state to another.

The assumptions here are that splenectomy can be done only for patients who are 5 years of age and above. Assuming that a person's life span is 100 years, the time state variable can take values from 5-100. After splenectomy is done, from then on, at every decision epoch, the time state variable is incremented by a value of 1 indicating the number of years that elapsed after the removal of spleen through splenectomy. Thus, the value of the time state variable depends on the value of the spleen state variable. If spleen shows a value of '1', that definitely is an indication that the time value is '0'.

4.1.3.5 Complications

This variable keeps track of any complication in a patient. This could be any type of complication, meaning any type of situation requiring immediate intervention. Complications can be of mainly two kinds.

- ω Complication occurring due to the presence of gallstones denoted by the complic (c) variable taking the value 1
- ω Complication ccuring due to the presence of spleen denoted by the complic (c) variable taking the value 2.

The condition of 'no complication' is denoted by a value zero taken by the complic (c) variable. Acute cholecystitis and biliary pancreatitis are examples of complications due to gallstones for which, an opensurgery may be required. Aplastic crisis is an example of complication occurring due to spleen, for which splenectomy is the remedy. The outcomes of the surgeries could be, that the patient is out of complication, implying that the value of the state variable 'complic (c)' is turned to '0'. Another outcome of the surgery could be surgical death, in which case also the 'c' variable takes the value of '0'. When, complication is present, the value assigned to 'c' is 1. If there is no complication in the current patient state, then the risk of occurrence of a complication in a future transition state is dependant on the level of the gallstone state variable. As the level of the gallstone variable increases, the probability of occurrence of complication increases. When the level of gallstone state variable is 5 or 6, the complication variable (c) takes the value '0' in the next state, since for gallstone at level 5, no complication can arise as the gallbladder has been removed. For gallstone in level 6 no complication can arise, as it indicates death. If the complication value in the current

patient state is '1', then a surgery is mandatory and the complication would have been

removed in the next transition state. Then, the transitioned state would definitely have a ‘c’ value of ‘0’, concerning that particular complication, concerning that particular complication.

4.1.3.6 Age

The decisions that a physician make might alter according to the age of a patient. Therefore, age is an important state variable. This state variable can take the values from 1 to100, assuming that patients can be in the range of one to hundred years old. After each decision epoch of the Markov chain, the age state variable is incremented by 1.

Age is not taken into consideration, as a state variable in the proposed simulation mechanism, due to the lack of knowledge of how the domain experts change their decisions depending on the age. But, the idea is that, age should be incorporated into the MDP modeling as a state variable, as it would differentiate the state of a patient depending on the age unlike the model of Paolo Magni et al.(2000)[8]. This variable if incorporated would contribute considerably to the state space of the system.

Though, age is not considered a state variable, it is taken into account in the present model, while dictating the optimal policy to the doctor who takes help of the decision support system. This is achieved by obtaining different optimal policies, according to the age of the patient. Usually, one optimal policy is obtained for a problem, but here, when age is taken into consideration, patients of different ages, become different optimization problems. The reason being, the maximum number of decision epochs, which can be traversed by different age patients are different. Though, the

methodology remains the same, it has to be applied, separately, to the different age patients to arrive at the respective optimal policies. Thus, age contributes to the decision making.

4.1.3.7 Sex

While resolving the tradeoff between the decisions to be taken, sex could be an important factor. Hence, it is appropriate to be added as a state variable denoting the patient state. This variable can take the values of '0' for male and '1' for female patients. However, the value of this variable remains the same through the decision process.

Sex also, has not been incorporated in the model due to lack of knowledge to approximate the behavior of the system based on this variable. In this thesis, sex has been taken into account in the model building process , but not while simulating the model.

4.2 Model solution

The solution to an MDP is called a policy and it simply specifies the best action to take for each of the states.

4.2.1 Simulation mechanism

The program 'Medical decision making' written in Java 2.0 programing language simulates a patient arrival and assigns a starting state to the patient. After an action is chosen, the patient goes to a transitioned state, from among a possible set of transition states. A reward or utility is generally assigned for that particular action, which is called the immediate reward. At the transitioned state, the decision maker again chooses an

action and the patient makes yet another state transition. This cycle continues until the patient dies or reaches the age of 100 after which the model assumes that patient is no more.

The states and the actions taken in those states, until the patient's death are noted. Also, the immediate rewards and the total expected reward are noted. Thereafter the model generates a new patient with a starting state and the cycle repeats.

The above-mentioned procedure is followed for a particular age group of patients. The optimal policy obtained, also pertains, only to this age patients. Thus, different optimal policies have to be obtained for different age patients. The methodology to obtain the optimal policy, though, remains the same.

4.2.1.1 Assignment of the starting state

Considering the present problem of HS, the starting states where a patient can begin the simulation, which corresponds to the situation of the patient when a doctor for the first time examines him/her, are found to be eleven. Equal probability is assigned for the patient to start in any of these 11 states. It should be recollected that the total state space is 2685 (including death states).

4.2.1.2 Input parameters

The action to be taken in a particular state is dictated by the reinforcement learning algorithm. When that action is performed in the respective state, the transitioned state to which the system moves is obtained by simulating that action in that state, in the simulation mechanism. This cycle repeats until the patient dies and the rewards are collected.

The numbers fed to the simulation program, which are the associated probabilities of going from one state to other, can be changed according to the user's knowledge pertaining to the information of his/her HS patients. These numbers could also be obtained from a medical database using tools like data mining or Bayesian learning. However, the methodology remains the same and the simulation-based reinforcement learning mechanism can work for any numbers, obtained in any manner.

The reinforcement learning algorithm developed in this research, to obtain the optimal policy, which maximizes the QALY's of a specific age patient, is presented next.

4.2.2 Average reward reinforcement learning

Here, the detailed steps of the algorithm are presented. This algorithm is a modified form of the algorithm given by Gosavi (1999) [46], adapted to the medical decision making problem considered, keeping in view the objective of maximizing the QALYs.

4.2.2.1 RL algorithm

1. Let the iteration count $m = 0$.

   Initialize a new patient arrival and assign *a* state*(s)* to the patient.

   Initialize action values $Q(s,a) = 0$ for all $s \in \xi$ and $a \in A(s)$, and the average reward $\rho_m = 0$,

   Initialize input parameters $(\alpha, \alpha_t)(\beta, \beta_t)(\gamma, \gamma_t)$,

   where, α represents learning rate,

   β represents average reward rate,

   γ represents the exploration rate.

2. While m < MaxSteps, do.

   If the system state at iteration m is $s \in \xi$,

   a) With a probability of $(1-\gamma_m)$, choose an action $a \in A(s)$ in state s, corresponding to the maximum $Q(s, a)$.

      Otherwise choose a random exploratory action from $\{A(s)\}$ with probability $\frac{\gamma_m}{(|A(s)|-1)}$.

   b) Simulate the chosen action a for the current state s.

      Let the system state at the $(m+1)^{th}$ decision epoch be s'. Let the immediate reward be $R(s, a, s')$.

   c) Update the $Q(s, a)$ value using the following equation

$$Q(s,a) \leftarrow (1-\alpha_m)Q(s,a) + \alpha_m[R(s,a,s') - \rho_m + Q_{\exp}(s')]. \qquad (4.9)$$

   d) Update the average reward ρ_{m+1} value as follows,

$$\rho_{m+1} \leftarrow (1-\beta_m)(\rho_m) + \beta_m\left[m \times \rho_m + \frac{R(s,a,s')}{m+1}\right]. \qquad (4.10)$$

   e) Update the learning parameters $\alpha_{m+1}, \beta_{m+1}$ and the exploration parameter γ_{m+1} following the Darken-Chang-Moody (1992) [47] scheme.

   For any parameter θ with θ_0 as its initial value and θ_t as the decay control parameter, updating is done as follows,

$$\theta_{m+1} = \frac{\theta_0}{1+u} \quad , \qquad (4.11)$$

$$\text{where,} \quad u = \frac{m^2}{(\theta_t + m)} \; . \qquad (4.12)$$

The elements α , β and γ are the starting values, and α_t, β_t, γ_t are large constants chosen suitably to control the learning and decay rates.

f) Set current state s to new state s', and $m \leftarrow m+1$.

3. If MaxSteps is reached, then go to step 4.

   Else, if s' is the death state, then initialize a new patient arrival having a starting state and go to step 2a.

   Else, go to step 2a.

4. Simulate the system with the final form of the Q-matrices to estimate the average value of the total QALY.

CHAPTER 5

NUMERICAL RESULTS

In this chapter, the numerical results obtained by applying the proposed solution methodology to the hereditary spherocytosis problem are presented. The solution methodology was tested with different values of the algorithm design parameters. The results presented represent the best solution.

5.1 Reinforcement methodology results

The solution methodology requires six design parameters. The design parameters are the initial values of the learning parameters (α , α_τ) for the Q-Values, learning parameters (β , β_τ) for the average reward ρ, and the exploration parameters (γ , γ_τ). The parameters α_τ, β_τ only affect the rate of decay of the corresponding learning parameters and are initialized to a large value of 10^{12}. The exploration decay parameter, γ_τ effects the rate at which exploration occurs and is initialized to 10^{11}. The average reward obtained from the RL methodology for various values of the exploration parameter (γ) and a fixed set of values for the learning parameters are listed in Table 2. The fixed values of the learning parameters (α , β) were obtained by trial and error.

Figure 8 shows a plot of the number of decision epochs versus the average reward obtained in each decision epoch during the learning phase of the RL methodology for different exploration parameter values, keeping the learning and the average reward learning parameters at a fixed value of 0.1.

Table 2. Results from the RL methodology

S. No	γ	Avg. QALY/year Learning Phase	Avg. Total QALY Learnt Phase
1	0	0.4356092	41.56283
2	0.1	0.4170014	41.37534
3	0.2	0.4346187	41.54974
4	0.3	0.444321	41.49474
5	0.4	0.4026054	41.15621
6	0.5	0.4007282	41.61528
7	0.6	0.390321	41.53495
8	0.7	0.4524216	41.95321
9	0.8	0.4392717	41.82469
10	0.9	0.4106183	41.32438

QALY = Quality adjusted life years

Note: All values are in generic units

Fixed values of the learning parameters (α , β) = (0.1, 0.1)

Fixed values of the learning decay parameters (α_τ , β_τ) = E12

Fixed value of the exploration decay parameter (γ_τ) = E11

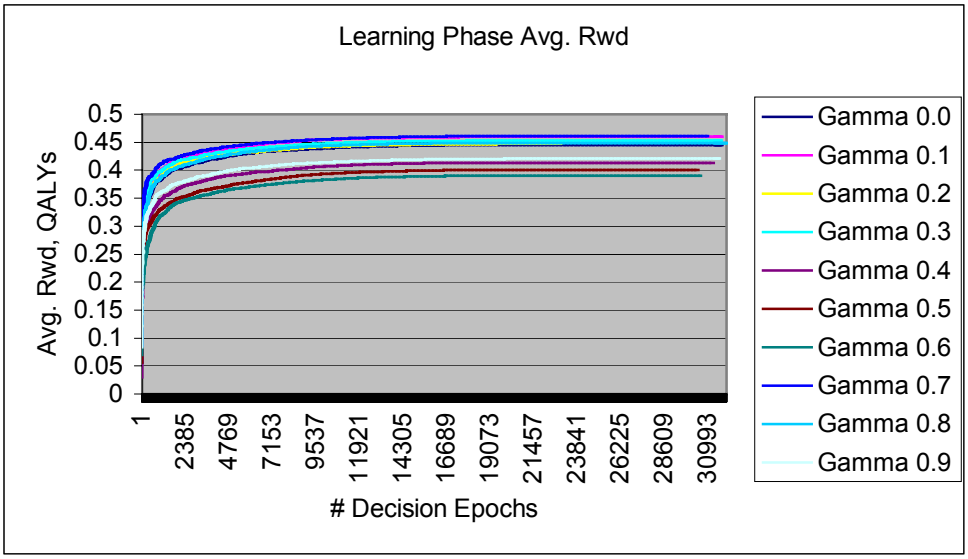

Figure 8. Average reward values for different exploration parameter values

The third column in Table 2 is the average reward in QALYs obtained for a patient in one year. This reward is obtained as a consequence of the decisions made during the learning phase of the RL methodology. The last column in Table 2 shows the average total QALYs per patient during his/her life time, which is the objective of the proposed methodology. These values are also called as the learnt phase values, because while obtaining these values, the RL methodology uses the best policy obtained in the learning phase. The corresponding combination of the design parameters associated with the highest learning phase average reward value would give the optimal solution. Thus, the highest average reward obtained during the learning, phase is 41.95321, corresponding to an $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 0.7$, α_τ & $\beta_\tau = E12$, $\gamma_\tau = E11$. Hence, it can be concluded that, the quality life that patients suffering from HS enjoy, would be, on an average, around 42 years, assuming that the patient lives for 100 years unless he/she encounters a surgical death or death due to side effects of a performed surgery.

5.2 Value iteration approach

According to Sutton and Barto [49], the term "Dynamic Programming" refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as an MDP.

Value Iteration is one such algorithm, which takes the transition probability matrices of different actions of the system and the reward matrix of the system as inputs, to compute the values of each state in the state space. Based on these values, the algorithm outputs a best policy. The best policy is a vector consisting of the best actions to be taken in the respective states such that the reward over the long run is maximized. [Please refer Appendix A for a description of the value iteration algorithm].

In the present problem, neither the transition probabilities nor the rewards are explicitly available. These have to be computed from the available information of the respective outcomes of the state variables, and their quality weights. Computation of transition probabilities from the known outcomes of various situations of a particular medical problem is part of the ongoing research on medical decision support. In the present thesis, a method is followed to obtain the transition probabilities and rewards from the known possible outcomes of the different levels of the state variables.
The method followed is explained below.

5.2.1 Method to obtain transition probability matrices (TPMs)

As mentioned earlier in section 4.1.1.5, there exists always a probability $P_{ss'}(a_i)$ of moving from state s to s' under action a_i. In the present problem, it should be noted that the states are characterized by the five state variables namely, gallstones(g),

spleen(s), Sepsis($\tilde{s}$), time(t) and complic (c). These variables further consists of levels. Therefore in the HS problem, there always exists a probability of moving from one state variable level to another level of the same state variable under action a_i, in one decision epoch, which is 1 year. These probabilities can be obtained from domain experts or abstracted from medical databases. In the present thesis, reasonable values are assumed for these probabilities for the five variables and are shown below. These are called variable transition probabilities from this point forward.

Table 3. State-variable transition probabilities in a decision epoch

Variable	Condition	Current level	Variable Transition Level	Variable transition Probability
Gallstone 'g'	Spleen = 0	1	1	1
		2	2	0.4
			3	0.4
			4	0.2
		3	3	0.6
			4	0.4
		4	4	1.0
		5	5	1.0
	Spleen = 1	1	1	0.2
			2	0.4
			3	0.3
			4	0.1
		2	2	0.2
Variable	Condition	Variable Current level	Variable Transition Level	Variable transition Probability
			3	0.5
			4	0.3
		3	3	0.3
			4	0.7

Table 2. (Continued)

Variable	Condition		Variable Current level	Variable Transition Level	Variable transition Probability
			4	4	1.0
			5	5	1.0
Complication 'c'	Spleen =0	Gallstone=1	0	0	1.0
		Gallstone=2	0	0	0.97
				1	0.30
		Gallstone=3	0	0	0.93
				1	0.07
		Gallstone=4	0	0	0.90
				1	0.10
		Gallstone=5	0	0	1.0
	Spleen=1	Gallstone=1	0	0	0.93
				2	0.07
		Gallstone=2	0	0	0.88
				1	0.05
				2	0.07
		Gallstone=3	0	0	0.83
				1	0.10
				2	0.07
		Gallstone=4	0	0	0.78
				1	0.15
				2	0.07
		Gallstone=5	0	0	0.93
				2	0.07
Sepsis '$\tilde{s}$'			1	1	1.0
	Spleen=1		0	0	1.0
	Spleen=0		0	1	5-(TimeValue-1)*0.0534

Table 2. (Continued)

Variable	Condition	Variable Current level	Variable Transition Level	Variable transition Probability	Variable
			0	0	100-[5-(TimeValue-1)*0.0534]
Spleen 's'			0	0	1.0
			1	1	1.0
Time 't'	Spleen=1		0	0	1.0
	Spleen=0		TimeValue	TimeValue+1	1.0

Note: All values are in generic units

These variable transition probabilities are the transition probabilities associated individually with each of the state variables. But the transition probabilities required for the value-iteration algorithm are the transition probabilities of moving from state '*s*' to state 's''. Therefore, a method is followed to get the state transition probabilities by grouping the variable transition probabilities. A patient state '*s*' is considered, which can be called the current patient state. All the possible states to which state '*s*' can transition, under a particular action (a_i) are figured out, depending on the state variable levels of the considered current patient state '*s*'. These possible states are called transition states. After that, the state variable levels of the current patient state are compared with the respective levels of the state variables of each transition state. The transition probability associated to transition from a particular state variable level of state 's' to a different level of the same state variable of a transitioned state 's'' is noted from Table 4, which has been called as the variable transition probability. Similarly, the transition probabilities for the other variables are also attained from Table 4. All these variable transition probabilities are summed up. Then another possible transition state is considered and the sum of the

variable transition probabilities for its variables is obtained. In this way, the sum of variable transition probabilities is obtained for all the possible states figured out. All these sums are again summed up, which can be called as the total sum. Out of the total sum, the percentage of the individual sums is calculated, which are the required transition probabilities from state s to all the possible transition states.

In this way for all the 2685 states, under different possible actions, the transition probabilities have been obtained and a transition probability matrix for each of the five actions has been developed. The rows of each transition probability matrix represent the probabilities of going from a particular state to all the other possible states in one transition or one decision epoch, for a particular action.

5.2.2 Method followed to obtain reward matrix

To develop the reward matrix also, a similar method is followed. A patient state '*s*' is considered and all the possible transition states ate figured out. The immediate rewards obtained for transitions to each of the possible states are calculated using the quality weights method given in section 4.1.2.11 and section 4.1.2.12. Then the immediate rewards are multiplied with the respective transition probabilities of the transition states. An average is taken over the products of the transition probabilities and immediate rewards, to obtain the reward in QALYs of taking that particular action in the state 's'. In this way, the immediate rewards obtained for all the states over all actions are put in a matrix form, which is the reward matrix. The rows of this matrix are the states and the columns are the five actions.

The value iteration algorithm uses these TPMs and the reward matrix to compute the actions which give the maximum value in each of the states. Sutton and Barto define value of a state or action, as a function which estimate how good it is for the agent (here, patient) to be in a given state or how good it is to perform a given action in a given state. Sutton and Barto furthur explain that "how good" refers to the future rewards the agent (here, the physician) can expect to receive in the future, which depends on what actions are to be taken. After computing the maximum value in each of the states, value iteration algorithm forms a policy, which consists of the actions corresponding to the maximum value in each of the states. But, this need not be the optimal policy. This could be one of the policies from the policy space. Therefore, the algorithm tries to improve the policy by calculating the values for each of the states again. In other words, it updates the values of the states using the below given update equation, which is another form of the Bellman optimality equation.

$$V_{new}(s) = \min_{a_i \in A(s)} \left\{ R_{sa_i} + \sum_{j=0}^{M} p(s, j, a_i) \times V_{old}(j) \right\}, \tag{5.1}$$

where, s is the current system state (here, patient state),

j is the transitioned system state,

M is the total number of states in the state space ξ of the system,

a_i is the action being considered,

R_{sa_i} is the immediate reward obtained for performing action a_i in state s, (whose value is obtained from the reward matrix).

$A(s)$ is the set of all actions possible in state s,

$p(s, j, a_i)$ is the transition probability to go from state 's' to state 'j' with action 'a_i'.

Theoretically, this updating of the values and improvement of the policy continues forever, requiring infinite number of iterations to converge to the exact optimal values and to obtain the optimal policy. But, in reality, the updating of the values and improving the policy is stopped after a finite number of iterations when the values change by only a small amount. The policy obtained is the optimal policy.

The average system reward by following the optimal policy obtained from the value iteration algorithm is 43.8790.

5.3 Policy differences

The difference between the value iteration technique and the proposed methodology is 1.21985, which is 1.22QALY's or 445.25days.The percentage difference between the two techniques is 2.825%. Part of the policies obtained by the help of these two techniques is given in Table 3 showing some of the differences between them. These differences partly contribute to the difference in the average reward obtained using them.

Table 4. Differences in policies of value iteration and reinforcement learning

State Position	State	Value Iteration Policy	RL Policy
384	(3, 0, 0, 3,0)	2	0
572	(3,0,1,2,0)	2	0
760	(4,0,0,1,0)	0	2
762	(4,0,0,2,0)	0	2
764	(4,0,0,3,0)	0	2
766	(4,0,0,4,0)	0	2
768	(4,0,0,5,0)	0	2
770	(4,0,0,6,0)	0	2
1520	(2,1,0,0,0)	2	3
1523	(3,1,0,0,0)	0	3
1526	(4,1,0,0,0)	0	2
1531	(5,1,0,0,0)	0	1

Note: All values are in generic units

CHAPTER 6

CONCLUSIONS

6.1 Concluding remarks

Medical decision making problems are typically characterized by a large number of different patient health conditions and many available treatment choices. Predicting the effect of a single treatment choice on the patient's health might not be difficult. But, predicting the effect of a chosen sequence of treatments, over the evolving health conditions of the patient with time, is perhaps impossible.

Medical decision problems often involve such sequential treatment strategies taken over a period of time. The objective of such treatment strategies involves choosing the best treatment from the available choice, in every health state of the patient such that, a preferred benefit measure is optimized. There is no existing framework to help analyze such sequential medical decision problems to obtain an efficient solution.

This thesis develops an efficient solution methodology for obtaining treatment strategies in sequential medical decision problems. The methodology involves modeling of the problems as a Markov decision process, and obtaining a solution using a reinforcement learning approach. Modeling as a Markov decision process accounts for the sequential nature of the problem, and the reinforcement learning based approach helps in obtaining a computationally efficient solution.

A medical intervention problem, Hereditary Spherocytosis (HS), with five treatment choices has been identified as a test bed to apply the methodology. In particular, after the physician has diagnosed a patient suffering from HS, the physician depending on the health condition of the patient tries to decide on a strategy, out of the possible strategies available in that particular health state. The benefit measure chosen, here, is the QALYs of a patient and the objective of the physician is to maximize the quality of life of the patient, over the patient's life. The solution obtained in terms of average total QALYs that can be obtained over a patient's lifetime has been compared with the optimal solution obtained from the value iteration algorithm of dynamic programming.

Experimental results using test data show that the proposed methodology can be effectively used to solve sequential medical decision problems with great reduction in computational effort when compared to the value iteration algorithms. Moreover, the optimal solution obtained by the proposed methodology was found to be quite close to that obtained using the value iteration algorithm of dynamic programming, hence giving a near optimal policy. The percentage difference, in the average total quality adjusted life years obtained per patient over the patients life, using the Value iteration technique and the reinforcement learning technique is found to be 2.825%. The difference being reasonable, it can be concluded that reinforcement learning is a viable alternative for the dynamic programming algorithms in obtaining a computationally effective solution. Moreover, reinforcement learning being a simulation-based methodology can be very useful in solving large-scale sequential decision problems in medicine.

6.2 Extensions to this work

Some of the extensions to this work are as follows,

- a reward scheme that accounts for cost of interventions and quantity of life along with the quality of life of the patient would make the model more realistic,
- development of a methodology to abstract the outcomes of the various events, which can also be called as transition probabilities, from a medical database using data mining tools,
- assumption that a patient lives for 100 years unless he encounters a surgical death or a death due to the side effects and complications due to certain treatment strategies can be relaxed to incorporate the natural death of the patient, which would be more realistic,
- accommodation of factors like age and patient while assigning quality weights,
- the issue that a patient being able to visit the physician whenever a problem arises, and the physician being able to take a treatment decision at any point of time, has not been incorporated in the present methodology. Therefore modeling the problems as a semi-Markov decision process to account for the changes occurring in the condition of a patient during a decision epoch, would considerably improve the model,
- patient states, usually, in a medical scenario cannot be defined perfectly as they are not fully observable. Therefore, modeling as a partially observable Markov decision process (POMDP) would get the model much nearer to the real life scenario.

REFERENCES

[1] Lin L., Poh K. L., Leong T. Y., Lim T. K., *Management Of Spontaneous Pneumothorax: A Decision Analysis,* In The 7th Congress of the Asian Pacific Society of Respiratory - APSR, Taipei, Taiwan, October 24-27, 2002, pp. 134.

[2] Leong T. Y., *Toward a general dynamic decision modeling language: An integrated framework for planning under uncertainty,* In Working Notes of the AAAI Spring Symposium on Decision Theoretic Planning, 1994.

[3] Lin L., Poh K. L., Lim T. K., *The Cost-Effectiveness Of Managing Chronic Cough,* In The 7th Congress of the Asian Pacific Society of Respiratory - APSR, Taipei, Taiwan, October 24-27, 2002, pp. 125.

[4] Harmanec D., Leong T. Y., Sundaresh S., Poh K. L., Yeo T. T., Nag I., and Lew T. W. K., *Decision analytic approach to severe head injury management*, Proceedings of the 1999 AMIA Annual Symposium, pp. 271-275, 1999.

[5] Zheng, J. and Leong, T. Y., *Consistency management in multiple perspective medical decision analysis,* In Proceedings of the 9th World Congress on Medical Informatics (MEDINFO98), 503-507, 1998.

[6] Paolo Magni, Riccardo Bellazzi, *DT-Planner: an environment for managing dynamic decision problems*, Computer Methods and programs in bio-medicine 54 (1997) 183-200.

[7] Marchetti M., Quaglini S., Barosi G., *Prophylactic splenectomy and cholecystectomy in mild hereditary spherocytosis: analyzing the decision in different clinical scenarios*, Journal of internal medicine 1998; 244: 217-226.

[8] Paolo Magni, Silvana Quaglini, Monia marchetti, Giovanni Barosi, *Deciding when to intervene: a Markov decision process approach*, International journal of medical informatics 60 (2000) 237-253.

[9] Hollenberg JP. *Markov cycle trees: a new representation for complex Markov processes,* Medical decision making. 1984; 4:529.

[10] Lau J, Kassirer JP, Pauker SG., *Decision Maker 3.0: improved decision analysis* by personal computer, Medical Decision Making. 1983;3:39-43

[11] Sonnenberg FA, Beck JR.; *Markov models in medical decision making; a practical guide,* Medical Decision Making, 1993; 13: 322-338.

[12] Beck JR, Pauker SG, *The Markov process in medical prognosis*, Medical Decision making 1984; 4: 529.

[13] Kassirer JP, Sonnenberg FA, *Decision analysis*, Textbook of Internal medicine, Philadelphia: J.B.Lippincott, 1988, 1991.

[14] Hazen G.B., *Stochastic Trees: A New Technique for Temporal Medical Decision Modeling,* Medical Decision Making, 12 (1992) 163-178.

[15] Hazen G.B., *Factored Stochastic Trees: A Tool for Solving Complex Temporal Medical Decision Models*, Medical Decision Making, 13 (1993), 227-236.

[16] Allan S.Detsky, Gary Naglie, Murray D.Krahn, David Naimark, Donald A.Redelmeier, *Primer on medical decision analysis: Part 1--Getting started*, Med Decis Making, 1997 Apr-Jun;17(2):123-5. Review.

[17] Allan S.Detsky, Gary Naglie, Murray D.Krahn, David Naimark, Donald A.Redelmeier, *Primer on medical decision analysis: Part 2--Building a tree*, Med Decis Making, 1997 Apr-Jun;17(2):126-35.

[18] Allan S.Detsky, Gary Naglie, Murray D.Krahn, David Naimark, Donald A.Redelmeier, *Primer on medical decision analysis: Part 3--Estimating probabilities and utilities*, Med Decis Making, 1997 Apr-Jun;17(2):136-41.

[19] Allan S.Detsky, Gary Naglie, Murray D.Krahn, David Naimark, Donald A. Redelmeier, *Primer on medical decision analysis: Part 4—Analyzing themodel and interpreting the results,* Med Decis Making, 1997 Apr-Jun;17(2):142-51. Review.

[20] Allan S.Detsky, Gary Naglie, Murray D.Krahn, David Naimark, Donald A.Redelmeier, *Primer on medical decision analysis: Part 5--Working with Markov processes,* Med Decis Making, 1997 Apr-Jun;17(2):152-9.

[21] Leong T. Y., *Representation requirements for supporting knowledge-based construction of decision models in medicine,* In Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care, pages 634-638, IEEE, 1991.

[22] Leong T. Y., *Representing context-sensitive knowledge in a network formalism: A preliminary report,* In Dubois, D., Wellman, M. P., D'Ambrosio, B. and Smets, P. (eds) Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference, pages 166-173, Morgan Kaufmann, 1992.

[23] Leong T. Y., *Dynamic decision modeling in medicine: A critique of existing techniques*, In Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care, pages 478-484, IEEE, 1993.

[24] Leong T. Y., *Toward a general dynamic decision modeling language: An integrated framework for planning under uncertainty,* In Working Notes of the AAAI Spring Symposium on Decision Theoretic Planning, 1994.

[25] Cao C. G. and Leong T. Y., *A learning approach to knowledge acquisition for supporting Markov therapy decision making,* In Working Notes of the AAAI Spring Symposium on Artificial Intelligence in Medicine: Applications of Current Technologies, pages 11-15, 1996.

[26] Leong T. Y., *An integrated approach to dynamic decision making under uncertainty*, TR-631, MIT Laboratory for Computer Science, August 1994.

[27] Leong T. Y., *A new methodology for clinical decision analysis over time: Theory and practice,* In Working Notes of the AAAI Spring Symposium on Artificial Intelligence in Medicine:Applications of Current Technologies, pages 89-93, 1996.

[28] Leong T. Y., *Multiple perspective dynamic decision modeling in medicine,* (abstract), In Proceedings of the Inaugural Conference of the Asia Pacific Association for Medical Informatics, 1994.

[29] Cao C. G., Leong T. Y., Leong A. P. K., and Seow F. C., *Induction of diagnostic test strategies with multi-level information measures,* Proceedings of Congress of International Medical Informatics Association (MEDINFO), pp. 477-482, 1998.

[30] Leong T. Y. and Cao C., Modeling medical decisions in DynaMoL: *A new general framework of dynamic decision analysis,* In Proceedings of the 9th World Congress on Medical Informatics (MEDINFO98), pages 483-487, 1998.

[31] Wang C. and Leong T. Y., *Knowledge-based formulation of dynamic decision models,* In Lee H. Y. and Motoda H. (eds) Topics in Artificial Intelligence: Proceedings of the 5th Pacific-Rim Conference on Artificial Intelligence (PRICAI98), pages 506-517, 1998.

[32] Leong T. Y., *Multiple perspective dynamic decision making,* Artificial Intelligence, 105:209-261, 1998.

[33] Sundaresh S., Leong T. Y., and Haddawy P., *Supporting multi-level multi-perspective dynamic decision making in medicine,* In Proceedings of the 1999 AMIA Annual Fall Symposium, pages 161-165, AMIA, 1999.

[34] Harmanec D., Leong T. Y., Sundaresh S., Poh K. L., Yeo T. T., Ng, I., and Lew T. W. K., *Decision analytic approach to severe head injury management,* In Proceedings of the 1999 AMIA Annual Fall Symposium, pages 271-275, AMIA, 1999.

[35] Peter Lucas, Ameen Abu-Hanna, *Prognostic methods in medicine*, Artificial Intelligence in Medicine, (1999) 15: 105-119.

[36] Qi X. Z. and Leong T. Y., *Constructing Influence Views from Data to Support Dynamic Decision Making in Medicine*, Proceedings of Congress of International Medical Informatics Association (MEDINFO), 2001.

[37] Cao C. and Leong T. Y., *Learning Conditional Probabilities for Dynamic Influence Structures in Medical Decision Models*, In Proceedings of the 1997 AMIA Annual Fall Symposium(formerly SCAMC), AMIA, 1997.

[38] Lin L., Poh K. L., Lim T. K., *The Cost-Effectiveness Of Managing Chronic Cough*, In The 7th Congress of the Asian Pacific Society of Respiratory - APSR, Taipei, Taiwan, October 24-27, 2002, pp. 125.

[39] Lin L., Poh K. L., Leong T. Y., Lim T. K., *Management Of Spontaneous Pneumothorax: A Decision Analysis*, In The 7th Congress of the Asian Pacific Society of Respiratory - APSR, Taipei, Taiwan, October 24-27, 2002, pp. 134.

[40] Xiang Y. P. and Poh K. L., *Knowledge-based Time-Critical Dynamic Decision Modelling*, Journal of the Operational Research Society 53(1):79-87, January 2002.

[41] Cao C., Leong T. Y., Leong A. P. K., and Seow F. C., *Dynamic decision analysis in medicine: A data driven approach,* International Journal of Medical Informatics, 51(1):13-28, 1998.

[42] Lau A. H. and Leong T. Y., *PROBES: A framework for probabilities elicitation from experts*, Proceedings of the 1999 AMIA Annual Symposium, pp. 301-305, 1999.

[43] Zhao F. and Leong T. Y., *A data preprocessing framework for supporting probability-learning in dynamic decision modeling in medicine*, Proceedings of the 2000 AMIA Annual Symposium, pp. 933-937, 2000.

[44] Dechter R., *Bucket elimination: a unifying framework for probabilistic inference*, UAI'96 Proceedings, Portland, OR, 1996.

[45] Marchetti M., Quaglini S., Barosi G., *Prophylactic splenectomy and cholecystectomy in mild hereditary spherocytosis: analyzing the decision in different clinical scenarios*, Journal of internal medicine 1998; 244: 217-226.

[46] Gosavi, *An algorithm for solving semi-Markov decision problems using Reinforcement Learning: Convergence analysis and Numerical Results*, Ph.D. Dissertation.

[47] Darken C., Chang J. and Moody J., *Learning rate schedules for faster stochastic gradient search*, In Proc. Neural Networks for signal processing 2. IEEE Press, 1992.

[48] Puterman M.L., *Markov Decision Processes*, Wiley Interscience, NewYork, 1994.

[49] Sutton R., *Reinforcement Learning*, Special Issue Of Machine Learning Journal, 1992.

[50] Littman M. L., Kaelbling L. P. and Moore A. W., *Reinforcement learning: A survey*, Journal of Artificial Intelligence Research, 4,1996.

[51] Mahadevan S., Marchalleck N., Das T. K. and Gosavi A., *Solving semi-markov decision problems using average reward reinforcement learning*, Management Science, 45(4): 560-574, 1999.

[52] Joshua Graff Zivin, *Health Valuation and environmental policy: A role for QALYs?,* March 2002.

[53] Bernard M. S. van Praag and Ada Ferrer-I-Carbonell, *Age-differentiated QALY Losses,* 23 April 2001.

APPENDICES

Appendix A: MARKOV DECISION PROCESS

A Markov decision process is a stochastic process characterized by 5 elements, namely, decision epochs, states, actions, transition probabilities and rewards. Also, there may be an agent (decision maker) that controls the path of the stochastic process. At certain points of time in the path, this agent intervenes and takes decisions, which affect the course of the future path. These points are called decision epochs and the decisions are called actions. At each decision epoch, the system occupies a decision making state. A vector that uniquely characterizes the system may describe this state. As a result of taking an action in a state, the decision maker receives a reward (which may be positive or negative) and the system goes to the next decision-making state with certain probability called the one-step transition probability. In a Markov process, the future state of the system depends only on the current state and the action chosen in the current state. A decision rule is a function for selecting an action in each state while a policy is a collection of such decision rules over the state space. A more formal definition of MDP is given next.

Sequential decision making problems, that are completely characterized by Markov chains as their only underlying stochastic processes, are commonly referred to in the literature as MDPs. Let,

$$X = \{X_n : n \in N, X_n \in \xi\} \tag{A.1}$$

Appendix A: (Continued)

denote the underlying Markov chain of an MDP, where, X_n denotes the system state at the n^{th} decision making epoch, ξ denotes the state space, and N denotes the set of integers. At any decision making epoch n, where, $X_n = i \in \xi$, the action taken is $A_n = a \in A_i$. A_i denotes the set of possible actions in state i and $\bigcup A_i = A$. Associated with any action $a \in A$ is a transition probability matrix P(a) of the Markov chain X, where $P_{ij}(a)$ represents the probability of moving from state i to j under action a. A reward function is defined as r: $\xi \times A \rightarrow R$, where, R denotes the real line, and r (i,a) is the expected reward for taking action a in state i. It is assumed that the rewards are bounded, rewards and the transition probabilities are stationary, and the state space is finite. Also, for the sake of simplicity, markov chains that are aperiodic and unichain are only considered.

The solution algorithms for MDPs, such as policy and value iteration, find the optimal stationary deterministic policy π^* (which is a mapping $\pi^* : \xi \rightarrow A$) that maximizes the reward criterion. A stationary deterministic policy refers to a policy that is independent of time. The Bellman optimality equation, which lies at the heart of dynamic programming methods like policy and value iteration algorithms, is stated next after defining two important terms gain and bias.

The gain for an MDP is defined as the average reward per period for a system in steady state under a given policy. When the system starts at any arbitrary state i and there after follows policy π, gain is given as

$$\rho^{\pi} = \lim_{N \to \infty} \frac{1}{N} E_i^{\pi} \left\{ \sum_{n=1}^{N} r(X_N, A_N) \right\} = \varphi r \ , \qquad \text{(A.2)}$$

Appendix A: (Continued)

where, φ denotes the limiting probability of the Markov chain X, and r is the reward vector {r(i,a) : i $\in \xi$, a $\in$ A}. The bias is defined as the expected total difference between the reward and average reward. Hence the bias in an MDP starting at state i and subsequently following policy π is given as

$$h^{\pi}(i) = E_i^{\pi}\left\{\sum_{n=1}^{\infty}\left[r(X_N, A_N) - \rho^{\pi}\right]\right\}. \tag{A.3}$$

A.1 Bellman optimality equation for average reward MDP's

Under considerations of average cost for an infinite time horizon for any finite unichain MDP, there exists a scalar ρ^* and a value function R^* satisfying the following system of equations for all i $\in \xi$,

$$R^*(i) = \max_{a \in A_i}\left(r(i,a) - \rho^* + \sum_{j \in \xi} p(i,a,j)\, R^*(j)\right), \tag{A.4}$$

such that the greedy policy π^* formed by selecting actions that maximize the right hand side of the above equation is average reward optimal, where r(i,a) is the expected immediate reward in state i, when an action a is taken, and p (i, a, j) is the probability of transition from state i to state j, under action a, in one state.

The average reward value iteration algorithm, which is one of many algorithms available for solving MDPs is given next.

A.2 The average reward value iteration algorithm

The value iteration algorithm is a method to iteratively obtain the optimal value function and the corresponding optimal policy using the bellman optimality equation. The

Appendix A: (Continued)

average reward version of the value iteration algorithm for MDPs (Puterman, 1994) [48], is presented next.

Let R_m (i) be the total expected value of evolving through m stages starting at state i $\in \xi$, and ψ is the space of bounded real valued functions on ξ.

- Select $R^0 \in \psi$, specify $\delta > 0$ and set m = 0 and a state $k^* \in \xi$
- For each i $\in \xi$, compute R^{m+1} (i) by

$$R^{m+1}(i) = \max_{a \in A_i} \left\{ r(i,a) - R^m(K^*) + \sum_{j \in \xi} p(i,a,j) R^m(j) \right\}. \quad \text{(A.5)}$$

- If sp ($R^{m+1} - R^m$) < δ, go to step 4. Otherwise increment m by 1 and return to step 2. sp denotes span, which for a vector ν is defined as span(ν) = max ν (i) – min ν (i).
- For each i $\in \xi$, choose the action d_δ (i) as

$$d_\delta(i) = \arg \max_{a \in A_i} \left\{ r(i,a) + \sum_{j \in \xi} p(i,a,j) R^m(j) \right\}, \quad \text{(A.6)}$$

and stop.

A value iteration sweep through the whole state space simultaneously updates the values in every iteration. This creates a considerable computational challenge, especially, for problems with large state space.

Even under favorable conditions, convergence of the average reward value iteration algorithm is very slow since V_n diverges linearly in n, becomes numerically unstable. A relative value iteration algorithm avoids this difficulty, but does not enhance

Appendix A: (Continued)

the rate of convergence. An asynchronous version of the relative value iteration avoids the sweep through the whole state space by updating the value of one state at a time. Such algorithms still require the complete knowledge of the system's probability structure and thus are difficult to implement for large systems. The computation of these quantities for problems with very large state spaces can become almost impossible. Hence, obtaining an optimal solution using these methods is often quite difficult.

In recent years, an alternative approach, called Reinforcement Learning (RL) that is based on simulation-based stochastic approximation has become a topic of intense research interest. Convergent algorithms based on this method have been shown to obtain near-optimal policies for Markov decision problems with a considerable reduction in computational effort. Reinforcement Learning algorithms have two distinct advantages over DP algorithms. The first advantage is that they can handle problems with complex reward and stochastic structures since they use simulation as a modeling tool. Secondly, RL can integrate within its various function approximation methods (regression, neural networks etc.), which makes it possible to solve problems with large state spaces.

Appendix B: REINFORCEMENT LEARNING

Reinforcement Learning (RL) is a way of teaching learning agents (decision makers) to predict the policy. This is accomplished by assigning rewards and punishments for their actions based on temporal feedback obtained during active interactions of learning agents with dynamic systems. Any learning model basically contains 4 elements, which are the environment, learning agents, and a set of actions for each agent and the environmental response (sensory input). Each learning agent selects an action and their actions collectively will lead the system along a unique path till the system encounters another decision making state. During this state transition, the agents gather sensory outputs from the environment, and from it, derives information about the new state and immediate reward. Using the information obtained during the state transition in conjunction with the algorithm, the agent updates its knowledge base and selects the next action. As this process repeats, the learning continues to improve its performance. A reinforcement-learning model is depicted in Figure 9. The learning agent provides the environment (system) with actions, and in return receives the sensory inputs that determine the next state and the reward or punishment resulting from its most recent action. On the n^{th} step of interaction, based on the system state $x_n = i$ and the reinforcement values $R(i) = \{R(i,a) : a \in A_i\}$ for the a available actions, the agent takes an action a, where $R^*(i) = \max_a R(i,a)$. The system evolves stochastically in response to

Appendix B: (Continued)

the input state-action pair (i,a) and results in outputs concerning the next system state x_{n+1} and the reward (or punishment) r (x_n,x_{n+1}) obtained during the transition. These system

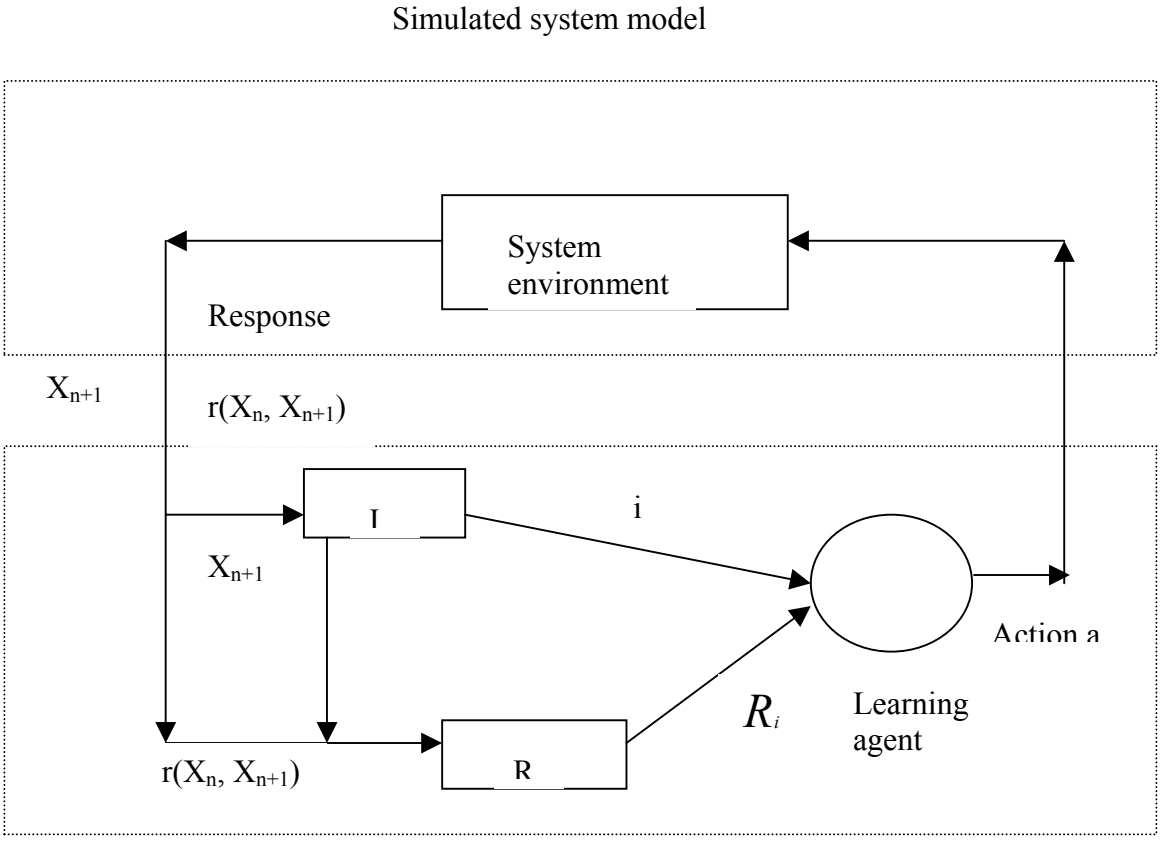

Figure 9. A reinforcement learning model

outputs serve as the sensory inputs for the agent. From these sensory data, the input function I helps the agent in perceiving the new system state.

Appendix B: (Continued)

Using the information about the new state (from I) and the sensory data about the reward (punishment), the reinforcement function R calculates the new action values R(i) for the previous state ($x_n = i$).

There are two different factors that determine the utility of an action. One is the immediate reward and other is the action value of the state to which the transition occurs as a result of that action. When a system visits a state, the decision maker chooses an action with highest (or lowest for minimization) action value (greedy policy). Initially, the action values for all state-action pairs are assigned arbitrary equal values (e.g., zeroes). When a system visits a state for the first time, a random action gets selected because all the action values are equal. As the system revisits the state, the learning agent selects the action based on the current action values, which are no longer equal. For ergodic processes, the states continue to be revisited and consequently the agent gets many opportunities to refine the action values and the corresponding decision making process. Sometimes, the learning agent chooses an action other than that suggested by the greedy policy. This is called exploration. As the good actions rewarded and bad actions punished over time, for every state, the action values of a smaller subset (one or more) of the actions tend to grow and others diminish. The learning phase ends when a clear trend appears with one or more actions in every state being dominant. These actions constitute the decision policy vector.

Appendix B: (Continued)

There are three different types of reinforcement learning models that have been studied most. In the finite horizon model, the agent optimizes the expected reward for a finite (h) number of steps,

which is given by

$$E\left(\sum_{n=0}^{h} r_n\right), \tag{B.1}$$

where r_n is the scalar reward received from the n^{th} step of the horizon. Hence, the agent's action on the first step is the h-step optimal action, on the second step h-1 step optimal action, and so on. The other two RL model types are infinite horizon models with average reward and discounted reward as their performance measures, which are given as

$$\lim_{h\to\infty} E\left(\frac{1}{h}\sum_{n=0}^{h}\gamma_n\right), \tag{B.2}$$

and

$$E\left(\sum_{n=0}^{\infty}\gamma^n r_n\right), \tag{B.3}$$

where, γ $(0 < \gamma < 1)$ is the discounting factor used per period. The concept of average reward is discussed briefly next.

B.1 Average reward RL

In most systems, the optimal total expected reward is finite either because of discounting or because of a reward-free termination state that the system eventually enters. In most situations, however, discounting is inappropriate and there is no natural

reward-free state. This makes it meaningful to optimize the average reward per stage starting from a state i, which is defined for any policy $\pi = (\pi_0, \pi_1, \pi_2, \ldots..)$ by

$$R^{\pi}(i) = \lim_{N \to \infty} \frac{1}{N} E\left(\sum_{k=0}^{N-1} (r(i,a,j) \mid i_0 = i) \right), \tag{B.4}$$

assuming that the limit exists, where r (i, a, j) is the reward received by taking action a in state i and going to state j.

B.2 Model based RL

One of the biggest difficulties encountered in solving MDPs with complex probability structures is to set up the transition probability matrices (TPM). If the TPM is available through mathematical calculations, one can always use classical methods like value iteration or policy iteration. Model-based RL usually computes the functions, such as transition probabilities and rewards using simulation. As the simulation progresses, the learning agent gets an improving estimate of these functions, and uses them in solution algorithms (e.g., Sutton, 1992) [49]. But the curse of dimensionality remains a problem with model-based RL. The ongoing research by the RL community is directed toward solving the dimensionality problem.

B.3 Model free RL

The model-based RL algorithms estimate the transition probabilities using simulation. Hence, a strong disadvantage of DP (i.e., the need for computation of

transition probabilities) is not avoided. The algorithms that obviate this need are referred to as model-free algorithms. Model-free algorithms can infer R-values directly from sample paths generated by simulation. For problems with large state spaces, the R- values need to be represented by some standard function approximator, such as a feed forward neural network, or a nearest neighbor Kernel regression algorithm.

Model-free algorithms belong to a class of stochastic iterative algorithms, of which a usual updating scheme for action values can be described as follows. Suppose that when an action a is chosen in state i, it results in an immediate reward of r_{imm}(i,a) and a system transition to state j. Then, the action value for the state-action pair (i,a) is updated as follows.

$$R_{new}(i,a) = (1-\alpha) R_{old}(i,a) + \alpha\left[r_{imm}(i,a) + \widetilde{R}(i,a)\right], \tag{B.5}$$

where, α is the learning rate, and $\widetilde{R}$(i,a) is an estimate of R (i,a) calculated from the feedback obtained during the system simulation. The exact form of $\widetilde{R}$(i,a) depends on the algorithm chosen and also on the performance metric. Q-learning and R-learning (Kaelbling et al., 1996) [50], SMART (Das et al., 1999) [51], RELAXED-SMART (Gosavi, 1999) [46] are all examples of model-free RL.

B.4 RL and DP

The relationship between DP and RL, which has its foundation in the DP framework, is discussed here. RL uses an interactive style of learning to obtain the optimal actions through trial and error.

Appendix B: (Continued)

The algorithms that drive the learning agent use the so-called reinforcement values that are actually related to the value function in DP are given below.

$$J(i) = max_a R(i,a) \qquad \forall i, \quad a \in A(i), \tag{B.6}$$

where, J (i) is the value function for state i, R(i,a) is the reinforcement value of taking action a in state i, and A (i) is the set of actions available in state i. RL calculates the reinforcement values (action values) for each state-action pair iteratively (using the well known Bellman equation) whenever a state-action pair is visited by simulating the system. DP, on the other hand, iterates over the reinforcement values of each state-action pair using the Bellman equation and pre-calculated transition probability and reward values. Hence, the primary difference between RL and DP is that RL stochastically approximates the system evolution through its state-action pairs, and DP considers random but stationary system state-action evolution.

B.5 RL and temporal difference methods

Here, the concept of temporal differences with reference to RL is discussed. The concept of temporal differences (TD) is central to the development of all algorithms in RL whether model-based or model-free. In this section, the following notational convention is used. For any given trajectory $i_0, i_1, \ldots, i_N$, with $i_N = 0$, and policy $\pi = (\pi_0, \pi_1, \ldots)$, let r (i, π_i, j) be the reward obtained by going from state i to state j under action π_i. Also, let $i_k = 0$, for k > N, and also r (i_k, π_k, i_{k+1}) = 0 for k >= N. It is assumed further that for any value function vector $R^{\pi}(.)$, $R^{\pi}(0)$ is zero.

Appendix B: (Continued)

For a trajectory ($i_0, i_1, \ldots, i_N$) that is generated, the reward estimates (value function) $R^{\pi}(i_k)$, k=0,….,N-1, can be updated as follows,

$$R^{\pi}(i_k) \leftarrow R^{\pi}(i_k) + \gamma(i_k)(r(i_k, \pi_k, i_{k+1}) + (r(i_{k+1}, \pi_{k+1}, i_{k+2}) + \ldots\ldots + (r(i_{N-1}, \pi_{N-1}, i_N) - R^{\pi}(i_k)). \quad \text{(B.7)}$$

Note that the above equation is actually the first step of policy evaluation in policy iteration methods. The update formula can be rewritten, for $R(i_N) = 0$, as follows,

$$R^{\pi}(i_k) \leftarrow R^{\pi}(i_k) + \gamma(r(i_k, \pi_k, i_{k+1}) + R^{\pi}(i_{k+1}) - R^{\pi}(i_k)) + (r(i_{k+1}, \pi_{k+1}, i_{k+2}) + R^{\pi}(i_{k+2}) - R^{\pi}(i_{k+1}))\ldots\ldots + (r(i_{N-1}, \pi_{N-1}, i_N) + R^{\pi}(i_N) - R^{\pi}(i_{N-1})) \quad \text{(B.8)}$$

The above equation is equivalent to Sutton's TD (1) update and can be expressed as

$$R^{\pi}(i_k) \leftarrow R^{\pi}(i_k) + \gamma(d_k + d_{k+1} + \ldots + d_{N-1}), \quad \text{(B.9)}$$

where, d_k denotes the k^{th} temporal difference and is given by

$$d_k = r(i_k, \pi_k, i_{k+1}) + R^{\pi}(i_{k+1}) - R^{\pi}(i_k). \quad \text{(B.10)}$$

The temporal difference d_k represents the difference between an estimate of the value function ($r(i_k, \pi_k, i_{k+1}) + R^{\pi}(i_{k+1})$) based on the simulated outcome of the current stage, and the current estimate $R^{\pi}(i_k)$. Thus the temporal difference provides an indication as to whether the current estimates R(i) should be raised or lowered.