A NEURAL NETWORK APPROACH TO TREATMENT OPTIMIZATION

by

Siripun Sanguansintukul

B.S., Chulalongkorn University, Thailand, 1988

M.S., Michigan Technological University, USA, 1993

Submitted to the Graduate Faculty of

School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Siripun Sanguansintukul

It was defended on

September 5, 2003

and approved by

Peter Brusilovsky, Assistant Professor, School of Information Sciences

Marek Druzdzel, Associate Professor, School of Information Sciences

Satish Iyengar, Professor, Statistics Department

Bambang Parmanto, Assistant Professor, Health Information Management

Paul Munro, Associate Professor, School of Information Sciences
Dissertation Advisor

A NEURAL NETWORK APPROACH TO TREATMENT OPTIMIZATION

Siripun Sanguansintukul, PhD

University of Pittsburgh, 2003

An approach for optimizing medical treatment as a function of measurable patient data is analyzed using a two-network system. The two-network approach is inspired by the field of control systems: one network, called a patient model (PM), is used to predict the outcome of the treatment, while the other, called a treatment network (TN), is used to optimize the predicted outcome. The system is tested with a variety of functions: one objective criterion (with and without interaction between treatments) and multi-objective criteria (with and without interaction between treatments). Data are generated using a simple Gaussian function for some studies and with functions derived from the medical literature for other studies. The experimental results can be summarized as follows: 1) the relative importance of symptoms can be adjusted by applying different coefficient weights in the PM objective functions. Different coefficients are employed to modulate the tradeoffs in symptoms. Higher coefficients in the cost function result in higher accuracy. 2) Different coefficients are applied to the objective functions of the TN when both objective functions are quadratic, the experimental results suggest that the higher the coefficient the better the symptom. 3) The simulation results of training the TN with a quadratic cost function and a quartic cost function indicate the threshold-like behavior in the quartic cost function when the dose is in the neighborhood of the threshold. 4) In general, the network illustrates a better performance than the quadratic model. However, the network encounters a local minima problem. Ultimately, the results indicate a proof of idea that this framework might be a useful tool for augmenting a clinician's decision in selecting dose strengths for an individual patient need.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my heart-felt thanks to those people who touched my life in so many different ways. Words cannot truly describe my experience and how I feel, because without their help, support and encouragement, I would not have been able to accomplish this research work. Although the task overwhelms me, I will try my best.

I am deeply grateful to my advisor, Paul, for his insightful suggestions, brilliant ideas, and problem solving. His goal was always nothing less than quality research. I admire his ingenuity and wish I had even half of his intelligence. His compassion and understanding about my personal life made it possible for me to do my work and still stay in touch with my distant family.

I would like to thank Satish for the statistical advice given to me during the experiments as well as his effort in reading and providing me with valuable comments on my papers.

I would like to thank Bambang for his help in so many ways in trying to find real medical data for the experiments. I have so many thanks for his friendship and understanding.

I would like to thank Peter for his moral support. His compliments of my GSA work make me feel proud as well as encouraged to do even better, whether it be for GSA work or my own.

I would like to thank Marek for his moral support and words of wisdom. He teaches not only the material in class, but also many aspects of the graduate student's life and beyond.

I truly thank my parents, from the bottom of my heart, for their love, support, and sacrifice throughout this long journey. Their prayers were always there for me, and even the vast ocean could not lessen the strength they gave me. My sisters and brothers have shared their caring thoughts, all for which I am blessed.

The person deserving special mention is my husband Chris. I am forever indebted to his unconditional love, unfailing support, and willing sacrifice. He provided much so that I may do my research; without his relentless proofreading, this thesis might have taken years. He inspires me to overcome the odds I encounter and lifts my spirit whenever it falls. His love is beyond my comprehension.

And to all of those others who helped me, in whatever way, to make this study a success.

1. Introduction

"*Clinical medicine is an art as well as a science*." [Albert, 1988] This is a common response from physicians to the question of how they arrive at a diagnosis or a choice of treatment. The above statement presents a reflection on the nature of reasoning in clinical medicine. Albert further commented that, in the view of most physicians, the reasoning process is somewhat incomprehensible and it goes beyond any set of rules. "*It is associated with intuition and creativity that cannot be substituted by any automated procedure"* [Albert, 1988]. Hence, the ability to arrive at a correct conclusion about diagnoses and treatments is considered to be a great triumph.

Physicians have to perform clinical judgments even with incomplete or imprecise information. Covell, Uman and Manning [Covell, 1985] demonstrated that clinicians are often not able to identify the sources of information they should use during clinical practice. In a busy practice, many information needs go unsatisfied. Osherhoff and colleagues [Osherhoff, 1991] [Forsythe, 1992] observed participants during the clinical practice of an academic health center to identify and classify information needs in a practice. They pinpointed three components of '*comprehensive information needs*':

1) Currently satisfies information needs (information recognized as relevant and already known to the clinician)
2) Consciously recognized information needs (information recognized as important, but not known by the clinician).
3) Unrecognized information needs (information important for the clinician to solve a problem at hand, but not recognized as such by the clinician).

They further stated that physicians might encounter tremendous amounts of patient information along with a huge amount of medical research literatures. The critical problem is to reconcile all that information quickly and efficiently [Osherhoff, 1991].

Additionally, health care professionals are often faced with complicated choices as a consequence of the imperfection in clinical data and the uncertainty in treatment outcomes [Owens, 2000]. Therefore, exploring approaches that can help health care providers to deal with the uncertainty in clinical medicine is drawing a lot of attention from investigators. Hence, medical decision support systems (MDDS) have been developed as a method for overcoming arbitrary clinical decision-making [Wellman, 1989]. The definition of medical (clinical) decision support systems by Musen [2000] is "*any program designed to help health professionals make clinical decisions.*"

New emerging techniques to assist clinical tasks such as diagnosis and treatment suggestions are in high demand. Designing a system that enhances the quality of care is a challenging goal. Artificial neural networks (ANNs) present an alternative methodology. ANNs have many substantive applications such as optimization, pattern recognition, classification, and many others - both realized and yet to be realized. The ANN adapts easily with new environments by learning, and can deal with information that is noisy and incomplete, an important feature in clinical domains. Furthermore, ANNs support generalizations, as well as providing a framework for predicting and managing uncertainty. These features have motivated extensive research and developments in ANNs. Due to the prevalence of complexities in medical treatment, it would undoubtedly be of great benefit to provide physicians with such medical decision support systems.

1.1 Motivation and Research Focus

Medical decision support systems (MDDS) have been developed for overcoming arbitrary clinical reasoning [Wellman, 1989]. Some of the fruitful areas in medical decision support systems are selecting effective therapies and making accurate prognoses. In some situations, it is not so much necessary to arrive at an exact diagnosis, as it is to initiate proper treatment before an exact diagnosis is made [Weiner, 2000].

The results of so many successful medical decision support systems based on the ANN [Baxt, 1996][Mango, 1996][Kennedy, 1997] methodology provide great encouragement for exploring MDDS and ANNs in new medical domains. The proposed research endeavors to build such a system for the purpose of augmenting a physician's decision in selecting effective treatments for patients. The system will take these factors into account: a patient's symptoms, findings in the illness conditions, selected treatments, and the degree of uncertainty in patients and physicians. Hopefully, the system will greatly enhance the treatment methodology and ultimately improve the quality of health care.

Various optimal control models have been used to model the dynamic systems of disease such as [Iliadis et.al, 2000][Stengel et.al, 2000] differential equations. However, the proposed research focuses on system control using the distal learning technique suggested by Jordan and Rumelhart [1992]. The optimal control structure utilized here is similar to that employed in many control systems such as in robotics and aircrafts. In this work, the technique of inversion contributes to the problem of finding the optimal value. Normally, the term 'inverse problem' or 'inversion' in control systems refers to the problem of finding unknown quantity solutions based on effect values obtained from the observation. This is known as 'kinematics' in the field of robotics. At the opposite end of the scale, there is the 'direct problem' in which the solutions are acquired from the effects when all beginning parameters are known.

Under the premise that the optimal treatments can be achieved if the right dosage of drug is given to a patient, the drug dosage required is to be optimized using the information available about the patient's physiology, the treatments (drug dosages), and the disease states (symptoms). Since the treatments can have different effects on the different patients, the adjustment of drug dosage according to an individual patient's need should result in maximizing the therapeutic efficacy as well as minimizing the side effects of the treatment. The ultimate goal is to ensure that quality care is delivered and the therapeutic outcome improves.

2. Background

With the remarkable growth of knowledge in medicine, together with the expectation of high quality health care services, physicians struggle to keep themselves current with the latest developments in the field. At the same time, physicians are not able to devote as much time for each patient case as in the past. Thus, most medical decisions require rapid judgments that rely on the physician's unaided memory. In spite of continuous training and re-certification procedures, human memory has limitations. In order to overcome these limitations and help physicians make diagnosis decisions quickly and effectively, endeavors have been made to build computer tools that assist physicians in storing vast quantities of information, retrieving appropriate knowledge, and most of all, giving suggestions for making appropriate diagnosis decisions [Szolovits, 1982].

"*Making the knowledge and expertise of human experts more widely available through computer consultation systems has been recognized as an important tool for improving the access to high quality health care*"[Szolovits, 1982]. The simulation of clinical cognition on the computer raises important scientific questions about the structure, consistency, completeness and uncertainty of medical knowledge. These considerations are of particular interest to researchers in artificial intelligence, cognitive psychology and medical information processing, and are important if we are to assess the performance and understand the role of computer consultation systems in medical education and practice [Szolovits, 1982].

Computer scientists, with the co-operation of health care professionals, have formulated a research discipline called Artificial Intelligence in Medicine (AIM). "*These researchers had a bold vision of the way AIM would revolutionize medicine and push forward the frontiers of* *technology*" [Coiera, 1997]. While AIM researchers aspire to develop tools that can enhance the efficiency of clinical decision-making and offer advice, these tools could never replace physicians. If the decision-making process is considered as more art than science, the computer will never be able to capture the "art" of skilled physicians [Harry, 1982].

2.1 Artificial Intelligence and Medicine

Without being comprehensive, the following section describes some of the early pioneering projects that led to many decision support systems present today. The emergence of AIM was in the early 1970s. Four of the earliest American research centers were: Stanford (MYCIN) [Shortliffe and Buchanan 1975, Shortliffe 1976], University of Pittsburgh (INTERNIST-1) [Pople, 1975][Miller, 1982], MIT (the Present Illness Program) [Pauker, 1977], and Rutgers (CASNET) [Weiss, 1978]. Some classic examples are presented below.

2.1.1 MYCIN

MYCIN is a decision-making system developed in the mid 1970s from a group at Stanford to identify the microorganisms that cause infections and give therapeutic advice. The name MYCIN is derived from the common suffix of many anti-microbial drugs. The aim of the program is to advise physicians and medical students on selecting appropriate treatment for an infection [Shortliffe 1975]. The MYCIN system was inspired from the earlier work, DENDRAL by Buchanan, Sutherland and Feigenbaum [Buchanan et.al., 1976]. DENDRAL is a rule-based system for chemical synthesis. MYCIN's knowledge was represented as production rules. A production rule is a conditional statement (IF…THEN…) that relates observations to the associated inferences. Some important characteristics of the MYCIN are:

1) It determined which rules to use and how to chain them together in order to arrive at a decision.
2) The system allowed the rules to be removed, altered or added without re-programming and re-structuring other parts of the knowledge base, allowing the knowledge structure to be modified rapidly.
3) Although rules were stored in machine-readable format, English translations could be displayed.

MYCIN has never been used clinically. However, the research and development of other knowledge-based systems during the 1970s evolved from this system. This includes EMYCIN, an outgrowth of MYCIN. EMYCIN, also known as "Essential MYCIN", is a domain-independent framework that can be used to build rule-based expert systems. EMYCIN has been used in a number of domains, both in medical (i.e. diagnosis consultant) and non-medical (i.e. engineering advisory system) domains.

2.1.2 INTERNIST

INTERNIST was developed at the School of Medicine, University of Pittsburgh, in the 1970s. It is a computerized decision support system in which the project goal was to model diagnoses in very broad areas of internal medicine.

INTERNIST combined database information with knowledge-based information to give consultative diagnostic advice. The INTERNIST database covered approximately 80% of the diagnoses of internal medicine [Pople 75]. Therefore, it is the largest of these AIM programs. There were nearly 600 diseases and 4500 interrelated manifestations. A manifestation or finding is a sign, symptom, or other patient characteristics that may be related to the diagnosis. On average, each disease was associated with 75-100 findings.

Physicians have realized that performing simple pattern matching would not be feasible since many findings, such as a fever, might be associated with multiple disease processes. In addition, estimating the conditional probabilities for all 600 diseases is impractical because many of those diseases are not well described in the literature. As a result, the developers resolved the problem by creating a scoring scheme to encode the relationships between diseases and the findings. The numbers frequency weight (FW) and evoking strength (ES) are assigned to each finding. These numbers indicate the relationship strength between the disease and the finding. The FW number ranges from 1 to 5, where 1 means the finding is seldom seen and 5 means the finding is always seen in the disease. Similarly, the ES ranges from 0 to 5, where 0 means the finding alone is not considered as the cause of the disease and 5 means the finding is the cause (pathognomonic) of the disease. With these weighting schemes and a simple knowledge structure, the INTERNIST system demonstrated impressive performance. However, the system was not widely used because of the significant cost of running the program and maintaining the database. The system can run only on mainframe computers. Later, INTERNIST grew into a decision support system named QMR (Quick Medical Reference) that can be run on the personal computers.

2.1.3 PIP (Present Illness Program)

PIP, or Present Illness Program [Pauker et.al, 1977] is a system that encodes the approaches that expert clinicians take while dealing with kidney disease patients. The system is able to diagnose and treat patients with acid/base and electrolyte disturbances. It contains about seventy frames of renal diseases. PIP used both categorical (or deterministic) reasoning and probabilistic (or evidential) reasoning. The PIP encompasses a large set of possible findings and hypotheses. The findings are facts about the patient, while the hypotheses are the program's assumption about the disease, clinical or physiological state from which the patient suffers. In addition, these hypotheses are associated with sets of prototypical findings that can either support or refute them. PIP's belief in a hypothesis is reinforced when findings reported from users are matched with these prototypical findings.

2.1.4 CASNET

CASNET, a diagnostic and therapeutic program for glaucoma and related diseases of the eye, was developed at Rutgers University by Kulikowski, Weiss and their colleagues [1978]. Causality was the essential concept of the diagnosis process in this system. The network is composed of a set of nodes; some are designed as starting states, while some are designed as final states. The causal relationships are represented by a link between two nodes. However, no cycles are permitted in the network. The developers believed that any abnormal phenomena have causal pathways that can be traced back to the factor-cause of the disease, either directly or from known antecedent states. When compared with the domain of renal disease of PIP or internal medicine of INTERNIST, the domain of this glaucoma system is much narrower and is well understood in physiological detail. The ONET (Ophthalmological Network), an organization which was founded to promote and test computer programs for Ophthalmological clinical research, estimated that CASNET '*arrives at reasonable and often sophisticated judgments in about 75% of the difficult cases of glaucoma*'[http://medg.lcs.mit.edu/ftp/psz/AIM82/ch2.html].

2.2 Selected Methods in Medical Decision Support Systems

This section presents selected techniques that can be applied in medical decision support systems.

2.2.1 Bayesian Networks

Bayesian networks are also called belief networks, causal probabilistic networks, and influence diagrams. Bayesian networks are the combination of graph theory and probability theory. The underlying graph theory involves the arrangement of nodes and edges in a graph. Graph models can be undirected or directed. Directed models can encode deterministic relationships such as an arc from A to B, which is interpreted as A 'causes' B. The directed graphical models are frequently used in statistical and AI communities. Besides the graph structure of the directed models, each node is associated with a conditional probability distribution.

A Bayesian network is a directed acyclic graph consisting of nodes, which represent random variables, and arcs, which represent direct probabilistic dependence between the variables they connect. The graphical representation of a Bayesian network reflects dependencies and independencies among the variables nodes. For instance, a variable is dependent upon its parents as well as its children. Even so, a variable is considered conditionally independent of a non-descendant variable if both variables have the same parent node(s). For instance, the Bayesian network in Figure 1 shows Heart Disease and Cancer, both of which have Smoking as a parent. Nevertheless, Cancer is conditionally independent of Heart disease.

Since heart disease and cancer are conditionally independent of each other, then the following is valid:

$$\mathrm{P}(\mathit{Heart\_Disease} \mid \mathit{Smoking}, \mathit{Cancer}) = \mathrm{P}(\mathit{Heart\_Disease} \mid \mathit{Smoking})$$

This means that the probability of heart disease is conditioned only upon smoking, and the value of cancer is irrelevant for this local probability. Similarly, it is valid to say that:

$$\mathrm{P}(\mathit{Cancer} \mid \mathit{Heart\_Disease}, \mathit{Smoking}) = \mathrm{P}(\mathit{Cancer} \mid \mathit{Smoking})$$

In this example, the joint probability distribution of smoking, heart disease, and cancer variables is:

$$\mathrm{P}(\mathit{Heart\_Disease}, \mathit{Smoking}, \mathit{Cancer}) = \mathrm{P}(\mathit{Heart\_Disease} \mid \mathit{Smoking}) \cdot \mathrm{P}(\mathit{Smoking}) \cdot \mathrm{P}(\mathit{Cancer} \mid \mathit{Smoking})$$

Examples of dependency relationships among different variables indicated from figure 1 are: *heart disease* depends on *smoking*; *cancer* depends on *smoking*; *survival* depends on *heart disease* and *cancer*; given *heart disease* and *cancer*, *survival* is independent of *smoking*; given *smoking* and *survival*, *heart disease* depends on *cancer*; and finally, with only *smoking*, *heart disease* and *cancer* are independent.

The Bayesian network approach has been utilized in applications such as primary bone tumor diagnosis [Kahn et.al, 2001], mammography interpretation [Burnside et.al, 2000], and liver disorder diagnosis [Onisko et.al, 1999].

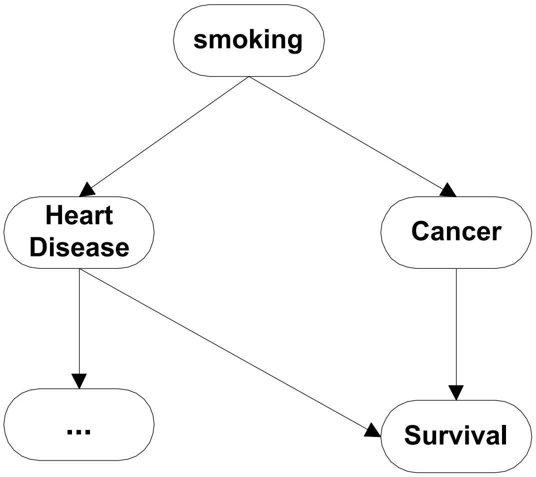


Figure 1: Bayesian Networks [Lucas, 1999]

2.2.2 Decision Trees

A decision tree is a graphical tool that expresses decision-making in a sequential fashion. Such a tree offers an explicit representation of decisions and outcomes, which are ordered according to the progression of time. Generally, the root of the tree is drawn on the left while leaves are drawn to the right. Together they represent a hierarchical set of rules describing how an object of interest might be evaluated or classified based on the answers to a series of questions. Each path in the tree, from its root to a leaf, represents a potential course of events and actions. As a result, the hierarchical rule structure of decision trees is easy to understand.

Figure 2 illustrates an example of a decision tree. The rectangular node represents a decision variable, while the circle node corresponds to a chance variable. Here, the decision variable *therapy* is associated with two possible actions: *surgical* or *medical* therapy. Each action is associated with the outcome of a chance variable *cure*, this denotes whether the patients are cured of the disease (yes – cured, no – not cured). Moreover, each chance is associated with a probability conditioned on the variables in the path from the root to that associated node. For instance, the number 0.1 represents the conditional probability P (cure=no | therapy=surgical). The values (i.e. 20 years, 2 years, 15 years) on the leaves of the decision tree are the utility function values. In this case, these values correspond to life expectancy. Using the rules of decision theory, it is expected to choose the therapy that maximizes life expectancy.

Example applications of decision tree methodology are diagnosis and management of dementia syndromes in primary care [Iliffe et.al. 2002], and diagnosis urinary tract infection [Goolsby, 2001].

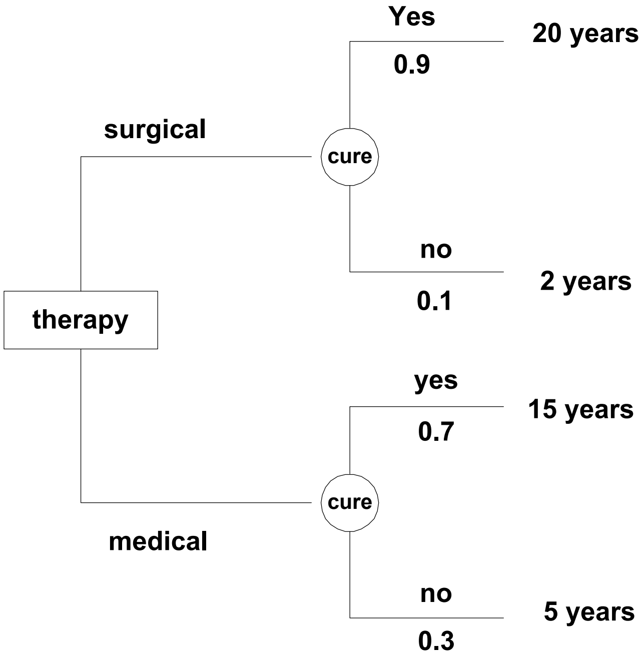


Figure 2: Decision Tree [Lucus, 1999]

2.2.3 Markov Process

The Markov system, also called the Markov process or Markov chain, is named after Andrei A. Markov, the first person to study the process systematically. A Markov system is a sequence of random events (states), each of which is correlated with the previous event (state). One state can move to another state, in each time step, with a fixed probability. For instance, a system in state *i* is moved to the next time step in state *j* with a probability P_{ij}. The probability P_{ij} is called *transition probability*. Moving from one state to another state is called *transition*. A Markov system can be illustrated by means of a transition diagram that shows all states, transitions, and transition probabilities in the system. An example of transition diagram for patient disease history can be seen in Figure 3. This diagram consists of nodes, which denote the states of the patient, and edges, which denote the transition between states. The number on the edges is the probability of the state transitions.

The Markov process has been employed in applications that model disease progression based on CD4 cell count for HIV application [Guihenneuc-Jouyaux et.al, 2000], and also for representing the amplitude of EEG in brain activity [Bai et.al., 2001].

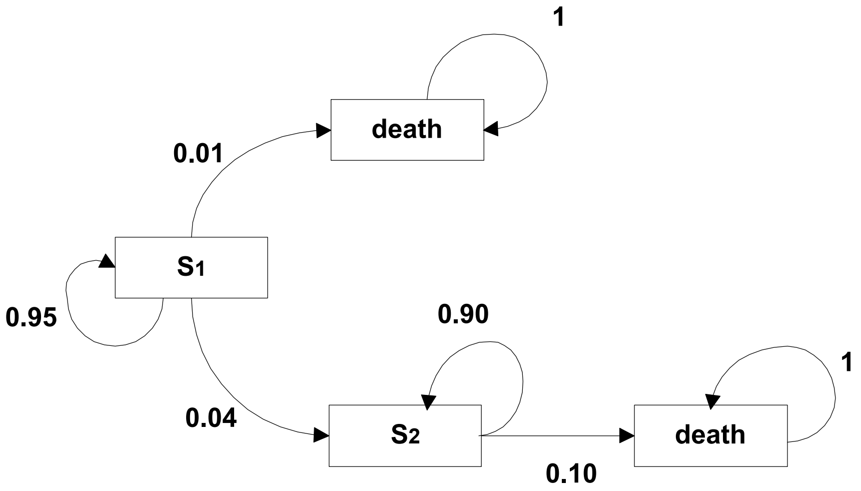


Figure 3: Markov Process [Lucas, 1999]

For instance, in health state S_1, a patient has a probability of 0.95 to remain in that state, a probability of 0.01 to die, and a probability of 0.04 to enter state S_2. This model can be simulated on a computer to calculate the average life expectancy of a number of patients. Furthermore, the probabilities can be easily modified to take into account the effects of aging on mortality, such as increasing the probability of death if a patient stays in a particular state for several time units.

2.2.4 Genetic Algorithms (GAs)

Genetic algorithms, based on natural biological evolution, were first introduced by John Holland [1975] during the 1970s. Thinking in terms of biological based evolution may facilitate the understanding of how genetic algorithms work. A brief explanation of the process is discussed here.

A set of possible solutions is randomly generated for a given problem. Those possible solutions are then evaluated according to a fitness level for each solution set. The parent solutions that are more fit have a better chance of generating successive offspring solutions than those that are less fit, so they are assigned a higher probability of survival. Generally, the populations are encoded as bit vectors and modeled after Darwinian evolution using operations such as crossover and mutation. The crossover operation usually takes two parents and generates two offspring. A random element position for splitting the parents is chosen, as can be seen in Figure 4, where the vertical bar denotes the splitting position. The simplest form of crossover uses a *one-point* crossover.

Parents	Offspring
(001010 | 1010)	(001010 | 0101)
(111001 | 0101)	(111001 | 1010)

Figure 4. One-point crossover

It is also possible that one or more bits (genes) in a solution will change over time. The changing of genes in a solution is called *mutation*. The simplest form of mutation is the random flipping of a bit. The resulting new generations will go through the same process as their parents until a solution is reached. Solutions essentially evolve over time. Typically, the average and maximum fitness increase over generations. An example of genetic algorithms in medical can be illustrated in Figure 5.

A hypothesis generally is represented by a string of bits. For example, the hypothesis *H1* in Figure 5, which is denoted by the bit string ‘1 01 010’, can be translated as:

IF (age > 65) ^
(1.5 mm < Tumor Thickness) ^
(Tumor Thickness <= 2.4 mm)

THEN Life Expectancy = 7 years.

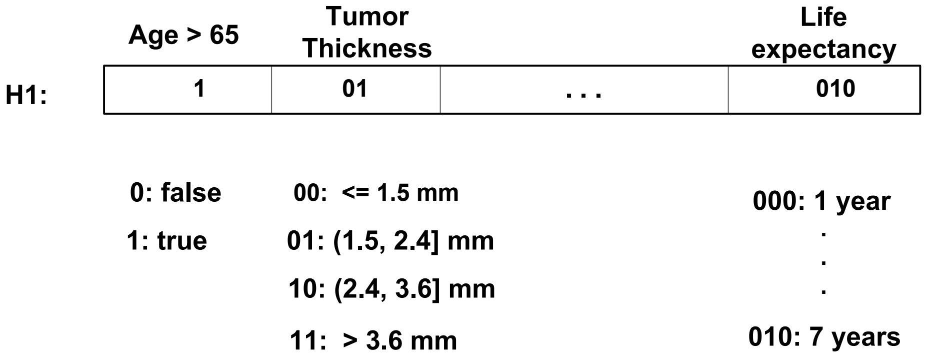


Figure 5: Genetic Algorithms [Lucas, 1999]

The genetic algorithm approach has been used, for example, in multi-disorder diagnosis [Vinterbo et.al, 2000], in selecting variables in the domain of myocardial infarction [Vinterbo etl.al, 1999] and in designing classifiers for lesion characterization [Sahiner et.al, 1998].

2.2.5 Artificial Neural Networks (ANNs)

An artificial neural network is a computational model that takes input, which encodes the information about patients (i.e. age, blood test, symptom), and outputs a number that can be further interpreted (i.e. the survival probability of patient). Learning in ANNs is based on the manipulation of the value edges weight W_{ij} so that the overall error of the predicted value of the outcome and the expected value is minimized. More information about ANNs in medicine can be found in [William, 1995][Penny, 1996][Tewari, 1997][Armoni, 1998]. The following figure demonstrates a three-layer network to predict the outcome based on the input values.

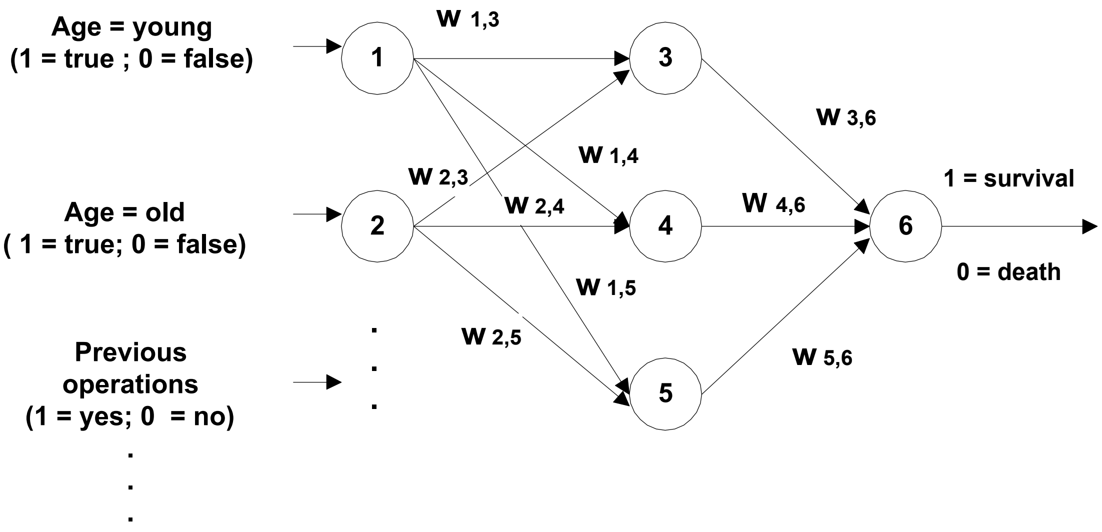


Figure 6: Artificial Neural Network [Lucas, 1999]

2.3 A Brief Introduction to Artificial Neural Networks

ANNs are also known as connectionist systems or parallel distributed processing models. They are non-linear, computational, mathematical models inspired by biological neurons. They were first developed in the 1960s, but became greatly popular in the mid 1980s after Rumelhart[1986] developed the back-propagation algorithm. Since then, the field continues to grow and there are a wide variety of network architectures and learning rules that have been explored.

Figure 7 illustrates the internal computation of a single neuron. The neuron is represented with a circle. Each input to the neuron, which is denoted by X_1, X_2,…, X_m, is multiplied by its corresponding weight factor W_1,W_2,…,W_m. These weight factors reflect the excitatory and inhibitory strength of the connection from the input source to the neuron. The sum of weight and input then goes through an activation function. A common type of activation function is logistic. It is frequently preferred because its derivative is continuous. The activation function also behaves like a switch. For instance, if the weighted sum of inputs exceeds 0.5 the neuron will fire or send out an appreciable output signal.

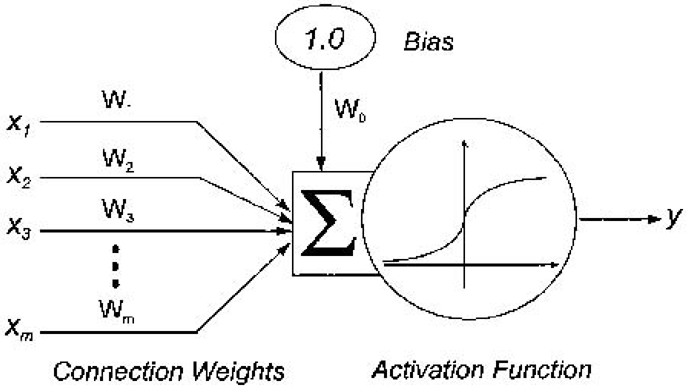


Figure 7: A single neuron [William, 1995]

There are many different types of networks varying in node model and pattern of connections [Looney, 1997][Mehrata, 1997][Reed, 1998]. However, the most common ANN in clinical medicine is called a multi-layer perceptron (MLP). In a MLP, the neurons are organized into three layers: input, hidden and output. Figure 8 shows the architecture of a MLP with 4 input, 3 hidden and 2 output nodes. Note that the terms *neuron* and *node* are used interchangeably. The output of a neuron in one layer is directed as input to every neuron in the immediately following layer. There are no lateral connections between neurons in the same layer. A MLP needs to have at least one layer of hidden neurons, which plays a critical role in performing the task. A MLP is considered to be a universal *approximator* [Hornik, 1989]. This means that any mapping function can be approximated with a single hidden layer given a sufficient number of hidden nodes. This includes all Boolean functions and any smooth non-linear function.

The connection weights are initially assigned with small random values. The network adapts its mapping function (of the input vector to output vectors pairs) according to the information presented. This process is performed using a training algorithm; the most well known such algorithm being known as *back-propagation*. As the learning progresses, the weights will be adjusted to reduce the error between the value in actual output and target output nodes. The process continues until the overall error becomes less than the pre-determined acceptable level resulting from the considered clinical scenarios.

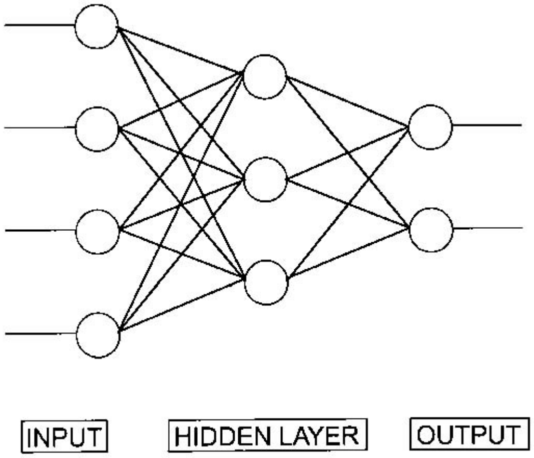


Figure 8: Multi Layer Perceptron (three layers) [Armoni, 1998]

2.3.1 Back-propagation networks

Back-propagation [Rumelhart, 1986] is a general-purpose learning algorithm based upon a form of optimization called *gradient descent*. Although there are many variations of back-propagation, the most basic type of back-propagation is most widely used [Penny, 1996]. The detail of this learning method can be found in [Hertz et.al, 1991][Werbos, 1994][Reed et.al, 1998]. In summary, the network is trained using data for which inputs, as well as desired outputs, are known. The input patterns are presented to the input layer and propagated through the network until they reach the output layer. In this forward pass, error signals, which are the error difference between the actual and desired outputs, are produced. These error signals are then propagated back to the network. The connection weights are adjusted to minimize the mean squared error of the error signals. Once the network is trained, the network weights are frozen and can be used to compute the output values for new input patterns.

Back-propagation can be used for classification, modeling, and prediction. For instance, in medical domain tasks, this learning algorithm has been used to classify patients with or without myocardial infarction [Baxt, 1991,1996][Kennedy, 1997] as well as to predict the cancer outcome [Burke, 1997].

Developing ANNs for specific applications involves topology selections (i.e., number of input units, number of output units, number of hidden units, number of hidden layers, connection between units), learning rule selections (i.e. mechanism for adjusting the weight), and possibly some other parameters such as learning rate and momentum.

Even though neural network based decision support systems appear to be highly accurate, many physicians hesitate to use this black box technology since the reasoning behind the computer's judgments are not explicit. In other words, the explanation of the ANN's decisions is not transparent. Nevertheless, Penny [1997] claimed "*there should be no problem of having faith in the black box since clinicians already do in everyday practice. Clinicians often make intuitive judgments without the ability to describe explicitly their thought processes to others. Therefore, ANNs should be judged on their performance the same as any other new clinical technology*".

Although ANNs will never replace human experts, they can assist in screening and can be used by clinicians to double-check their diagnosis. Meaningful benchmarks for performance measurement must be included before ANNs become established as reliable and practical tools in medical decision support systems.

2.3.2 The Potential Positive Impact of ANNs in Health Care

The quality of health care is becoming significant because of the intense competition among the health care delivery systems. As a result, the health care market demands to have objective measurements for comparing physicians and hospitals across health care organizations. ANNs in medical decision support appear to be one of the promising approaches to improving the quality of the health care delivery systems.

Many significant roles of ANNs in medical decision support are [Reggia, 1993] [Hirshberg, 1997][Lisboa, 1999][Tafeit, 1999]:

1) Enhancing the consistency of care.
2) Potentially covering rare conditions, since it is impossible for clinicians to possess the knowledge of all diseases, especially in exceptionally rare cases.
3) Exhibiting adaptation or learning, which allows the system to become intelligent and generalize to cases that the system has never before seen.
4) Eliminating issues associated with human fatigue, emotional state and working conditions.
5) Enabling analysis of conditions and diagnoses in real time.

6) Enabling rapid identification of known cases.

It is known that when clinical data are imperfect and outcomes of treatment are uncertain, health professionals often are faced with complicated choices [Owens & Sox, 2000]. Exploring an approach that can help health care providers to deal with the uncertainty inherent in medical decision support systems is drawing a lot of attention from investigators. Among the several approaches for automating diagnoses, rule-based systems and the Bayesian approach seem to be the most widespread. Rule-based systems typically require formulation of explicit diagnosis rules. Unfortunately, much of human knowledge, especially medical expert knowledge, remains implicit. Implicit knowledge is generally difficult to formulate, or even express verbally. Additionally, translation of implicit knowledge into explicit rules tends to cause distortion or loss of information content. In the Bayesian approach, symptoms of diseases are assumed to be statistically independent. This assumption does not hold in the reality of diagnostic practice [Miller, 1994]

ANNs present an alternative methodology because they can be trained to correlate the symptom data with associated diagnoses to predict the diagnosis outcome. Furthermore, ANNs support generalizations; this is important in clinical domains that have incomplete and noisy data. ANNs also provide a framework for predicting and managing uncertainty.

ANNs have much to offer modern medicine. To name a few applications, ANNs that diagnose heart attacks or myocardial infraction received publicity in the Wall Street Journal [Pacific Northwest Laboratory, 2000]. ANNs showed superior diagnosis accuracy to physicians [Baxt, 1991;1996][Kennedy, 1997]. This application is especially important in the emergency room where physicians have to handle large amounts of data and the critical decisions have to be made within time constraints. Another example is PAPNET [Mango, 1996], which is a commercial product that employs ANN technology in the diagnosis of cervical cancer. This product is proven to increase the efficiency of manual screening pap smears. Furthermore, an ANN’s ability is not affected by factors such as fatigue, emotional state, and working conditions. ANNs are effective in solving clinical problems because of their properties in experience-based learning, fault tolerance, graceful degradation, and signal enhancement.

Medical decision support systems ensure that basic care is not overlooked so that physicians can concentrate on more sensitive issues. In the case of certain diagnoses, the systems can recommend appropriate therapeutic procedures in order to ensure the standard of care for that particular disease. In the case of coexisting conditions such as when patients appear to have concurrent symptoms, which deviate from the standard treatment protocol, treatment decisions may become very complicated [Weiner, 2000].

2.4 Distal learning

Multilayer neural networks (Rumelhart, Hinton and Williams, 1986) have attracted a great deal of attention due their potential applications in a variety of fields such as engineering and medicine. Feed-forward multilayer networks are the most frequently used since they have been applied successfully in a wide range of tasks.

Learning algorithms in neural network fall into two categories with respect to the source of feedback to the network (learner); namely, unsupervised learning, and supervised learning. In unsupervised learning, the learner receives no feedback from the environment. The task of the learner is to represent the inputs in a more efficient way as clusters or categories based on the similarities or differences in the input patterns. In supervised learning, the learner is provided with the target; that is the learner is told what the response should be. Then, the learner learns to adjust its actual response to the target so that it can respond with the appropriate output the next time the same input is presented.

In many circumstances, however, it is not unusual that there are no teachers to provide direct information to the learner. For instance, a basketball player learns how to shoot baskets (example from Jordan and Rumelhart 1992). The challenge faced by the learner (the basketball player) is in determining what muscles must be applied in order to create the necessary physical movements that drive the balls into the goal successfully. The problem becomes even more complex when one considers many different goal positions and visual scenes, which in turn might involve different actions or different muscle commands.

The question is what learning algorithm could be applied to do the mapping from visual scenes to the muscle commands when the only target information available is the outcome of the actions i.e. whether balls passed through the baskets. Performing the clustering or feature extraction on the visual input is not sufficient; hence, unsupervised learning cannot be used. How about supervised learning? Classical supervised learning is not sufficient either, for the reason that, there is no teacher to provide the muscle commands as target. Remember that, the only information given to the learner is the result from the action movement.

The general scenario motivated by the basketball example is demonstrated as follows:

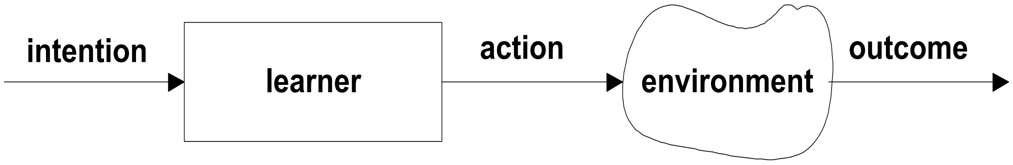


Figure 9: Distal supervised learning problem [Jordan & Rumelhart, 1992]

In this learning system (see Figure 9), the learner's inputs are *intentions*. The learner then transforms the intentions into *actions*. The actions are further transformed by the *environment* into *outcomes*. Actions (proximal variables) are directly controlled by intentions, while the outcomes (distal variables) are indirectly controlled by the intentions through the proximal variables. During the learning process, it is presumed that target values are available only for the distal, but not proximal, variables.

Essentially, the model can be understood from two different points of view: from the learner's point of view and from outside the learning system. From the view outside the system, the distal learning task is a mapping from the intentions to the outcomes. On the other hand, from the view of the learner, there is a mapping from the intentions to actions with the integration of the environment in order to produce desired distal outcomes. The teacher in these learning tasks is called *distal teacher*, meaning, far from the point of the origin. As mentioned before, the training information is available for the outcomes (distal variables) rather than the actions of the learner (proximal variables). This means that the learner has to determine how to adjust proximal actions to minimize the distal error. Utilizing the information obtained from interacting with the environment, the learner should be able to make appropriately adjustments from input-to-action.

The theoretical idea behind the above scenario is to expand the notion of classical supervised learning. To enhance the capabilities of classical supervised learning algorithms, the internal model has been suggested for modeling the environment. As a result, distal supervised learning can be applied to the broader range of applications. A similar idea has been proposed by several researchers, such as Williams [1986] and Munro [1987].

2.4.1 Supervised learning with a distal teacher

The following section describes how to implement the distal teacher technique using connectionist networks. The training is based on a two-phased procedure:

- The first phase, or forward model, performs outcome predictions associated with the context of the given state and actions. Specifically, this internal model transforms a given state and actions into distal outcomes. The learning is based on adjusting the error difference between the predicted and actual distal outcomes. Since the target values are provided directly to the forward model, the forward model itself is actually a supervised learning problem. Once the forward model has been partially trained, the weights will be frozen and employed in the next phase.

- The second phase, or learner model, employs the previous model that has been learned in order to indirectly solve the mapping from intentions to actions. This means that the learner must make the appropriate tuning, from inputs to actions, based on the data obtained from interacting with the environment. There are two networks connected into series: the learner network and the forward network. The forward network is the one that has been trained in the previous phase. This composite network can be treated as one supervised learning system. However, the weights in the forward model are held fixed for the reason that the system should focus solely on finding the optimal value by varying only the mapping from inputs to action.

In this phase, the forward model performs outcome predictions based on the consequence of actions chosen by the learner. At the end, the forward model propagates the error, the difference between the actual outcome prediction and the desired outcome, back to the learner model. This allows the learner to discover how to optimize the mapping from intentions to actions.

If the trained forward model is a good model, in that it is capable of determining the optimal solutions for the composite system as a whole, then the ability of the learner model to map inputs to actions should also become good.

In conclusion, distal supervised learning could be solved using a composite system that consists of a forward and learner model connected in series with respect to each other. It is necessary that the forward model be trained prior to the training of the composite system. By training the composite system to map from inputs to distal targets, this training process could indirectly solve the mapping from input to actions.

Jordan and Rumelhart have commented that the learning algorithm employed does not necessarily have to be back-propagation learning. Any supervised learning method can be used, provided that the first derivatives can be differentiated, to learn a mapping across composite networks, including a sub-network that has been previously trained.

2.4.2 Distal learning and inverse models

Distal learning paradigms were originally developed to solve problems in control systems. Control theorists, especially in adaptive controls, have devoted considerable attention to control problems involving distal teachers [Gullapali, 1992]. One such problem receiving extensive study is kinematics, which provides one of the clearest demonstrations of the use of the distal teacher.

A brief description of inverse kinematics is given as follows. Readers interested in more detail about inverse kinematics can also refer to [Demers, 1992].

The study of the relationship between the joint variables of a robot manipulator and the position and orientation of the end-effector is part of the kinematics discipline. [Craig 86, McCarthy 90].

The robot manipulator or robot arm is a collection of links interconnected by flexible joints. At the end of the robot, there is a tool or *end-effector*. The end-effector is the part of the manipulator that usually interacts with the environment.

Generally, if the values of joint variables are known, the location of end-effector can be easily calculated. The mapping from the joint space (i.e., angle of the joint variables) to the workspace (i.e., desired position of the end-effector) is called *forward* kinematics. Nevertheless, we usually encounter the inverse kinematics situation when we want to control the end-effector to reach a particular location. Therefore, we need to compute the joint variables so that the end-effector is placed at the desired location. The problem of finding inputs (a set of joint variables) that yield a desired output (end-effector target orientation and position) is called an *inverse* kinematics problem.

2.4.3 Other Related Works

Research work along the same line as distal supervised learning is called *network inversion*. Network inversion has proved to be a valuable tool in numerous engineering applications [Jensen and Reed, 1999].

A trained feed-forward network can be considered as the nonlinear mapping from the input space to the output space. The objective of training feed-forward networks is to determine the value of connection weights. Once the network is trained, the weights are held fixed. From this result, the mapping from the input to output space can be determined. On the other hand, the mapping of the feed-forward network from output to input space is referred to as *inverse mapping* or *inversion*. It is necessary to invert trained feed-forward networks in order to examine and improve the generalization performance of the trained networks [Lu, 1999]. Inverted networks play significant roles in solving many problems encountered in the fields of engineering and science.

Williams [1986] is the pioneer who proposed that networks could be inverted using the back-propagation learning algorithm. Linden and Kinderman [1989] employed network inversion as a tool to comprehend the behavior inside networks. The networks were implemented using the gradient descent search in input space. Here, the networks were trained to perform classification of handwritten digits from 0-9. Afterwards, some features of the trained networks were extracted and then compared with the typical example of each digit. It should be noted that their method is different from back-propagation training, which searches through the weight space.

Hoskins [1992] applied a network using an iterative constrained inversion technique as a method for making control inputs to an industrial plant. A straightforward method was used for training the forward model. Then an iterative inversion was performed to generate the control commands on-line. This approach allowed the controllers to dynamically respond to changes in the plant.

In the study of Behera and colleagues [1996], an inversion algorithm for radial basis function networks, based on an extended Kalman filter, were developed. The purpose of this algorithm was to perform the trajectory tracking control of a two-link manipulator. A feed-forward network was first trained to approximate the forward kinematics of the manipulator. Then the network was inverted to produce their desired goal.

Ishida and Zhan [1995] employed network inversion for the application of controlling the growth of a crystal. Since temperature is a very important factor for success in the growth process, maintaining a specific temperature is mandatory throughout. Typically, there are several electric heating elements in the growth chamber for controlling the temperature. The concern is what the heat rate should be so that the desired temperature is produced. To achieve this goal, the network was first trained to predict the temperature at time $t+1$ given the temperature and the heat rate of each heater at time t. Then, the network was inverted with the output held fixed at the desired temperature at time $t+1$ so that the required heat rate could be determined.

Davis and co-workers [1991] applied the improved network inversion technique for automated cytology screening. The focus was on using the inverted network to classify the boundary surface of cervical smear images, particularly Squamous Intraepithelial Lesions (SIL), from normal cells. Typically, a very large number of SIL slides contain only a low incidence of SIL cells. This means that number of normal cells and artifacts vastly outnumbers SIL cells in a SIL slide. Thus, the automated screener should work in a manner that it is capable of rejecting the artifacts and maintaining good cell sensitivity at the same time. The inputs were the features extracted from image segmentation and feature extraction stages before the classification process began. The extracted features consisted of 12 features, some of which are: area, perimeter, compactness, and nuclear-cytoplasm constract. The simulation results showed a significant improvement in the accuracy of classification.

Lu and colleagues [1995] demonstrated inverse kinematics for redundant manipulators. In their approach, the configuration space and associated workspace are partitioned into several regions. A set of modular networks were trained using forward kinematics on those regions. Those networks were then inverted to obtain the inverse kinematics solutions for a desired end-effector position. Finally, the optimal solution was chosen from the solutions on the previous step according to the given criteria. A comprehensive review of network inversion techniques can be found in [Jansen and Reed, 1999].

2.5 Optimization Techniques

The goal of an optimization problem is to acquire the independent variables or factors which optimize (maximize or minimize) the performance of the system, subject to allowed parameter ranges and some other restrictions. The performance quantity to be optimized is called the *objective function*. The restrictions on allowed parameter ranges are known as *constraints* and the parameters that can be changed in the search for an optimum are termed *decision variables*.

Notation of terminology used:

- **Decision variable**

A decision variable, or control variable, is any parameter that is being controlled in order to achieve an optimum. This parameter has an important impact in the behavior of the system being optimized. Decision variables are normally represented by a vector $x = \{x_1, x_2, .., x_n\}$. Decision variables can be continuous or discrete in function.

- **Objective function**

An objective function, also called a cost function, is denoted as function *f(x)*. This objective function can be used to measure the effectiveness of the design. An optimization problem can have one or multiple objective functions. The optimization that attempts to achieve multiple objectives at the same time is referred to as *multi-criteria optimization* or *multi-objective optimization*.

- **Constraint**

A constraint is an extent or condition that limits flexibility in varying the decision variables. The constraint can be formulated as either one of equality or inequality. A constraint equation in the form of $g_i(x) = 0$ is called an *equality* constraint, while, $g_i(x) \geq 0$ is an *inequality* constraint.

- **Standard form**

The general optimization problem can be expressed as:

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & g_i(x) = 0, \quad i = 1,2,...,m' & \text{[equality constraints]} \\ & g_i(x) \geq 0, \quad i = m'+1,...,n & \text{[inequality constraints]} \end{array}$$

This can be read as finding the decision variables x such that the function $f(x)$ is minimized and the given constraints $g_i(x) = 0$ and $g_i(x) \geq 0$ are satisfied. It should be noted that a minimized function is the same as the maximized negative of that function (A function can be maximized by multiplying the objective function with −1). For example, a maximum of function f is a minimum of $-f$. Therefore, it is sufficient to discuss only minimization problems.

The method for modeling and optimizing a system is known as mathematical programming. Mathematical programming can be divided into linear programming (LP) and nonlinear programming (NLP).

2.5.1 Linear Programming (LP)

The important characteristic of LP is that all mathematical functions (objective functions and the corresponding constraints) in the model are required to be linear functions. The word *programming* is essentially intended to mean *planning* rather than computer programming. For this reason, linear programming can be interpreted as the planning of activities in order to obtain an optimal result [Hillier and Liebermann, 1990] that best achieves the specified goal. The goal is to find a point that minimizes the objective function and at the same time satisfies the constraints. Any point that satisfies the constraints is referred to as a *feasible* point. In a linear programming problem, the objective function is linear, and the set of feasible points is determined by a set of linear equations and/or inequalities. The goal of LP is to determine the values of decision variables that maximize and minimize a linear objective function, where the decision variables are subject to linear constraints.

There are many application problems that can be modeled using LP. The most common problem is allocating limited resources among competing activities in the most optimal manner, such as scheduling crews for an airline flight. Besides this common problem, LP can be employed in various areas such as agricultural planning and radiation therapy.

A technique frequently used to solve LP problem is the *simplex* method [Dantzig, 1947]. This method was developed during World War II to find the best possible method for efficiently allocating numbers of armies, machines, and supplies. In short, '*the simplex method is an algebraic procedure, where each iteration involves solving a system of equations to obtain a new trial solution for the optimality test*.' [Hillier and Lieberman, 1990]. Since it is more convenient to deal with equations of equality than inequality, slack variables are introduced to convert inequality constraints into equality constraints. A more recent approach to solving LP problems that was invented in 1984 is called *Karmarkar's interior point method* [Karmarkar, 1984]. More information about these two methods can be found in [Nash et.al, 1996].

2.5.2 Nonlinear Programming (NLP)

Many real life applications are nonlinear problems. In general, NLP is concerned with the optimization of a function *f(x)*, subject to some constraints, where the objective function and/or corresponding constraints are nonlinear functions. NLP problems are broad and can be divided into many classes such as: quadratic programming, geometric programming, convex programming, unconstrained optimization, constrained optimization, etc. Different techniques have been developed to solve individual classes of these problems. This means that there is no single methodology that can solve all of these different types of problems. However, they can be broadly categorized into constrained and unconstrained optimization problems.

2.5.2.1 Constrained optimization

A constrained optimization problem refers to a problem in which the optimized function is restricted to the function of decision variables. A number of methods are capable of solving this type of problem; however, a classical method to deal with this problem is called the method of Lagrange multipliers. In this method, solutions are obtained by converting a constrained optimization problem into an unconstrained optimization problem. Consider a constrained problem derived from Fonner[1970], which can be mathematically formulated as follows:

$$\text{minimize} \quad y = f(x_i), \qquad i = 1,2,...,n \qquad [1]$$

$$\text{subject to} \quad g_j(x_i) = \alpha_j, \quad j = 1,2,..., p \leq n \qquad [2]$$

$$g_j(x_i) \geq \alpha_j, \quad j = p+1,...,m \qquad [3]$$

The objective function is presented in Equation [1]. This is the function that needs to be optimized (find the minimum or maximum). The equations [2] and [3] denote the equality and inequality constraints for any specified constants α_j.

The procedure starts by formulating the Lagrangian function. In order to do this, all inequality constraints must first be converted into equality constraints. Therefore, a slack variable q_i is introduced to equation 3. Furthermore, this slack variable must be non-negative so that its value is positive in the Lagrangian function. After introducing the slack variable into equation 3, the resulting equation becomes:

$$g_j(x_i) - q_i^2 = \alpha_j, \quad j = p+1,...,m \qquad [4]$$

The Lagrangian function is essentially the addition of the objective function, the constraints and the product of Lagrange multipliers(λ). The Lagrangian function of the above constrained problem (equation 1-3) is:

$$h(x,\lambda) = F = f(x_i) + \sum_{j=1}^{p} \lambda_j [g_j(x_i) - \alpha_j] + \sum_{j=p+1}^{m} \lambda_j [g_j(x_i) - q_i^2 - \alpha_j] \qquad [5]$$

The new variables $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ are called Lagrange multipliers. The numerical value of these variables can be used as a measurement to envision the increased or decreased value in the objective function corresponding to a unit of change in the constraint [Fonners, 1970]. Finally, solutions can be obtained by solving *n*+*m* equations simultaneously. Each equation is a partial derivative of the Lagrangian function related to X_i , λ_j and is set to 0 or: $\frac{\partial h}{\partial x_i} = 0$, $\frac{\partial h}{\partial \lambda_j} = 0$

2.5.2.2 Unconstrained optimization

An unconstrained optimization problem is concerned with finding the optimal value of an objective function without any constraints. For instance, if only equation [1] in the previous problem is considered, then this problem can be referred to as an unconstrained optimization problem. There are several methodologies to handle this problem, such as Newton-Raphson, steepest gradient, etc. However, the main focus here is the steepest descent.

- ***Newton-Raphson Method***

In short, the Newton-Raphson method is an iterative process for solving non-linear equations simultaneously. Essentially, it is part of the gradient method for optimizing unconstrained functions. Since it makes use of the second order derivatives, the cost of computation and storage is high. In addition, if the starting point for search is not close to the solution, then this method is not guaranteed to converge to a minimum of the function.

- ***Gradient method: Steepest descent***

A gradient is essentially a slope of the function at any point. Generally, the gradient of a function f is denoted by $\nabla f(x)$. The gradient method utilizes the information of the function's slope to search for a direction where a minimizer or maximizer might be located. It can be proved that function *f* has the maximum increase rate in the direction of the gradient. The discussion of the proof can be found in [Chong et.al 2001]. As a result, $\nabla f(x)$ points to the direction of $f(x)$ which has the maximum increase rate, while $-\nabla f(x)$ points to the direction of $f(x)$ that has the maximum decrease rate. In other words, to find a function minimizer, the direction of the negative gradient is a good direction to follow.

The gradient descent, or simply gradient algorithm, can be formulated as follows:

$$x^{(k+1)} = x^{(k)} - \alpha \ \nabla f(x^{(k)}) \qquad [6]$$

When $x^{(k)}$ is a given point, $x^{(k+1)}$ is the next point and $\alpha\nabla f(x^{(k)})$ is the amount of a move. The term α is known as a step size. The step size can be chosen as small or large. The gradient method is terminated when the gradient of a point becomes null. Gradient methods are widely employed in many applications because of its simplicity and performance.

Among the different methods applying the above idea is one called *steepest descent*. In particular with using the steepest descent, the step size α is chosen such that the objective function is decreased at the maximum for each step. Steepest descent is an iterative procedure. A number of points are generated from this method, with each point calculated based on the preceding point. In addition, each new generated point will cause the value of the objective function to be maximum decreased. Therefore, steepest descent is a descent method.

- ***Unconstrained optimization and feed forward neural networks***

To perform any function mapping, neural networks have to be trained. The most widely used training method is back-propagation. The back-propagation training algorithm can be formulated as an unconstrained optimization problem. Therefore, an optimization technique involving search methods such as steepest descent can be used to select weights in the network.

Lets assume that a multi-layer network strictly consists of three layers. Learning a task in multi-layer networks is analogous to adjusting all synaptic weights so that the error between the desired output and the actual output over all the training examples are minimized. Therefore, training the standard back-propagation using the steepest descent approach can be considered as the following unconstrained optimization.

$$\text{minimize } E = \sum_{p}\sum_{j} \ (d_{jp} - y_{jp})^2$$

The objective, or cost function, is the sum of the square error between the desired output d_{jp} and their corresponding output y_{jp} of the j-th output for the p-th pattern. More information about unconstrained optimization and neural networks can be found in [Cichocki et.al., 1993] and [Chong et.al, 2001]

To handle constrained optimization problems using neural networks, hard bounds can be introduced to limit the values in inputs or outputs to the network. Another alternative is combining another term, called a penalty function, to the cost/objective function. The values that lie outside the constraints or produce an unfeasible solution will get a penalty.

To deal with multi-criteria optimization problems using neural networks, several techniques can be employed. These include a weighting method, minimax method, L_p-norm method and ε-constraint method. The detail of each methodology can be found elsewhere [Cichocki et.al. 1993]. Here, the focus is on the weighting method. A scalar weight, which indicates a relative significance to other factors, is assigned to each concerned decision factor. The sum of all introduced scalar weights should be equal to 1.

2.6 Dose Optimization Review

Dose refers to a nominal measurement of drug exposure [Derendorf, et.al., 2000]. Dose takes many forms, such as: radiation dose, intake drug dose, and dose of components or ingredients in drug designs. When applying the concept of optimization into medical treatments that focus on dose, the objective function can be the health status of the patient. The constraints include, for example, limitation of drug concentration, limitation of toxicity level, level of white blood cells, and level of high blood pressure.

One of the major goals in clinical pharmacology is to identify an optimal dose and dose regimen for a patient with a disease state. This is based on the assumption that there is a causal relationship between the dose, dose regimen, medication exposure and the drug's action as far as both beneficial and adverse effects [Derendorf, et.al. 2000]. Two major disciplines in clinical pharmacology are known as pharmacokinetic (PK) and pharmacodynamic (PD). PK is "*the study of the movement of drugs in the body, including the processes of absorption, distribution, localization in tissues, biotransformation and excretion*" [WebMD]. PD is "*the study of the mechanisms of action of drugs and other biochemical and physiologic effects*" [WebMD]. The classical meaning for PK is "what the body does to the drug", while PD is "what the drug does to the body". It has been claimed recently that PK/PD modeling is involved in all stages of drug development. However, the great interest here is PK/PD modeling in optimizing dose recommendations. It is noteworthy to say that a number of researchers utilize underlying knowledge in PK/PD for treatment optimization.

The field of optimization methodologies in medicine is very broad. For brevity, this review will classify optimization procedures into two classes of techniques: analytic and heuristic. Analytic techniques endeavor to model essential feature mechanisms of the interested factors using mathematical equations and then attempt to solve those equations for optimal solutions. Heuristic techniques are based on an iterative adjustment process of concerned attributes until they produce satisfactory (if not optimal) solutions.

2.6.1 Analytic Technique

- ***Immune system and Chemotherapy***

Stengel and colleagues [2002] studied treatment of pathogenic disease processes. The goal of their study was to demonstrate how to apply a drug regimen so that it maximizes the efficacy while simultaneously minimizing the side effects and dollar cost. The authors interpret the disease process as an optimal control problem in a dynamic system. This disease model is generated using non-linear, fourth-order differential equations that describe essential features of the disease dynamics such as pathogens, plasma cells, antibodies and health status of the patients. The model studied here is a generic one; it does not represent any specific pathogen.

The cost function is defined as an addition of the terminal cost (concentration of a pathogen and organ health, at the end of treatment period) and an integral cost of state and control variations (concentration of a pathogen, organ health, pathogen killer, plasma cell enhancer, antibody enhancer, organ healing factors). In addition, a coefficient is introduced to each (squared) variable. These coefficients establish the relative importance factor in the treatment cost.

The conditions for optimization are formulated by three Euler-Lagrange equations. These equations are composed of a linear, ordinary-differential equation and a stationary condition. Finally, the steepest-descent is employed to solve this problem.

In cancer chemotherapy, the central concern is to design the drug regimen treatment strategy to ensure that the tumor cells are killed at the desired rate. At the same time, the dose given should not cause an unacceptable toxicity to the hosts (patients), since chemotherapy simultaneously kills tumor cells and triggers toxicity in normal cells. Hence, the optimal treatment can be achieved if a drug is given in the right dose. Mathematical ideas and models borrowed from optimal control theory have been used to reduce tumor burden and to keep side effects at acceptable levels. One of the first pioneers in this research area is Swan [1990].

Iliadis et.al. [2000] formulated a mathematical model to describe the pharmacokinetics and pharmacodynamics of anticancer drugs, antitumor efficacy, and drug toxicity. The constraints associated with this model include allowed plasma concentrations, drug exposure and white blood cell populations. The model specifies how to deliver the drug at the maximal rate under the specified toxic constraints. Specifically, this study considers drugs as single agents and models the regimen optimizations at the first cycle of chemotherapy

The model consists of a number of differential equations: five objective-functions and four constraints. Factors of interest in objective functions are: drug concentration in plasma, active drug concentration in plasma, number of tumor cells, white blood cell populations and drug exposure. The constraints consist of: limitation of maximum drug concentration, limitation of maximum drug exposure, level of white blood cells and maximum period that white blood cells remain in low level. Solutions of these equations are obtained either by integrating equations or numerically solving using Runge-Kutta method. The authors utilized non-linear programming and numerical integration techniques from Matlab to solve this problem.

- ***Radiation***

Typically, radiation can be delivered in two different modes: internal and external. The internal therapy refers to the implantation of radioactive sources directly into the cancer tissue. The external therapy uses an external beam source that shoots directly into the cancer tissue. External therapy will be discussed here.

One of the important issues in external therapy is the beam-boundary problem. Therefore, the treatment plans, which deal with the shape of radiation beams, time durations, etc., should be planned so that it is optimal in such a way that the dose distribution can destroy a tumor without a major risk to the surrounding healthy tissue or nearby major organs. In the conventional treatment planning technique, the dose planner assigns the number of treatment fields, weights, and shapes. However, if the dose is not satisfactory, the task is repeated and the process is restarted. This is called forward planning. On the other hand, if the desired dose on the patient is prescribed first and the treatment fields, shapes etc are computed later, this is called inverse planning. Both forward and inverse planning utilizes mathematical equations to model the relationship between the treatment fields and dose. Interest in inverse planning strategies has increased recently. The following figures (derived from http://venda.uku.fi/research/radiotherapy/invprob/invproblem.html) show both forward and inverse planning. In Figure 11: forward planning, dose distribution is computed according to given treatment settings (i.e. treatment fields, given direction, intensity profiles). In Figure 12: inverse planning, the organ is partitioned according to the dose-volume requirements and appropriate constraints are assigned to each segment.

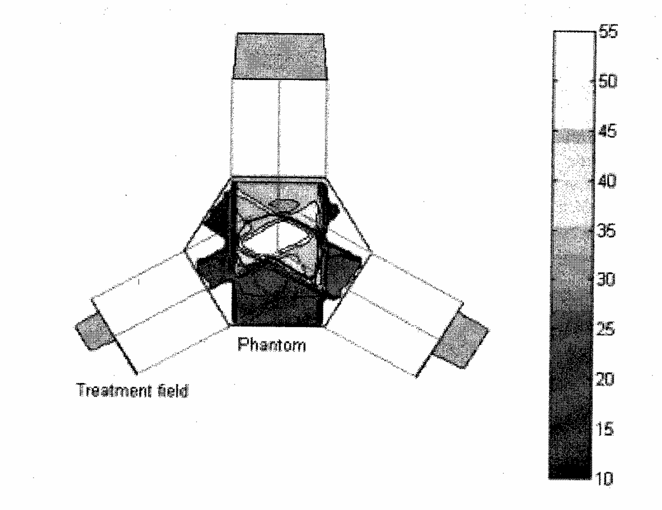


Figure 10: Radiation: forward planning
[http://venda.uku.fi/research/radiotherapy/invprob/invproblem.html]

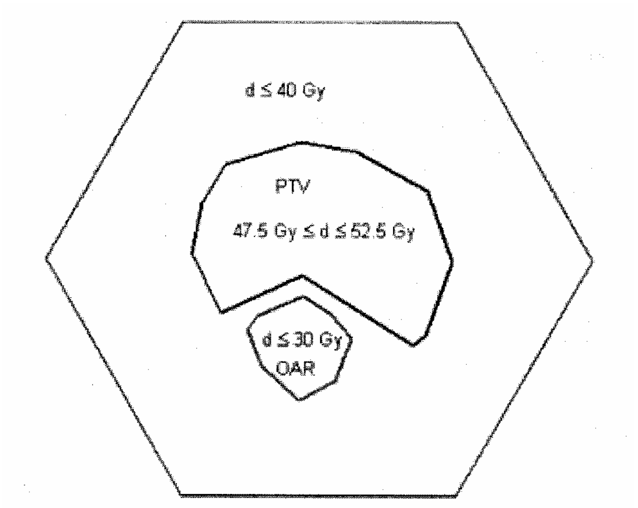


Figure 11: Radiation: Inverse Planning
[http://venda.uku.fi/research/radiotherapy/invprob/invproblem.html]

Cho and co-workers [1998] present two optimization models for intensity beam modulation with dose-volume constraints imposed on the bladder and rectum. One model is based on cost function minimization (CFM) while the other model uses the theory of projections onto convex sets (POCS).

In the CFM technique, the cost function is the sum of the target objective (the sum square difference between target dose and prescribed dose) and the penalty function. The penalty function is a multiplication of dose and volume penalty terms. "*The constraints are designed to penalize solutions that increase the fractional volume permitted to exceed a dose limit while maintaining the entire organ to below a maximum bound*" [Cho et.al.1998]. The solution could be obtained using existing programs such as simulated annealing.

In POCS, the non-convex problem of dose-volume constraints are re-formulated in terms of the integral dose limit, which permit the use of convex constructs. In addition, it does not form a total cost function to be minimized. The solution can be acquired "*through the process of orthogonal projections whose directions are determined by minimizing the Euclidean distance between the convex constraint sets. The projection operation is analogous to taking the gradient in the cost function based on iterative methods*"[Cho et.al, 1998].

Other optimization techniques related to intensity modulated beams can be found in Langer et.al.[1987],Mohan et.al.[1996] and Niemierko [1996].

- Pharmacology

Optimization methodologies have been developed for structuring problems in pharmaceutical research such as [Fonners et.al. 1970], one of the earlier researchers in pharmacological optimization (old but classical paper). A common problem in the dosage of pre-formulated products, for instance, is to optimize the product with one or more blend elements so that it has all the required characteristics such as effectiveness, safety, and usefulness.

To design the pharmaceutical products and pre-formulations, the pharmaceutical researchers often encounter problems such as attempting to identify the relationship between the independent variables and their corresponding responses. The problem becomes even more complicated when the response of one variable differs from another (i.e. one response tends to improve while another deteriorates). The desirable formulation for one property is not always desirable for other characteristics. In other words, a conflict might arise when two or more objective functions are required to be optimized at once. This is also called the multi-objective optimization problem.

Fonners et.al. [1970] established complicated pharmaceutical design problems by formulating them as constrained optimization problems. Their experiments involved finding the concentration level of stearic and starch that optimized the physical properties of the tablet in a drug called “Phenylpropanolamine hydrochloride”. The constraints were tablet hardness, friability, volume, in vitro release rate of the drug and urinary excretion rate of the drug in human subjects. The problems were solved using the Lagrangian method (see Section: 2.5.2.1Contrained Optimization Technique).

2.6.2 Heuristic Technique

One of the widely used techniques to achieve optimal product or process design under pre-specified constraints and interactive effects is called response surface methodology or RSM (will be discussed in section 2.7). This optimization technique is well documented in the literature

[Box and Draper, 1987]. Response surface methodology (RSM) consists of mathematical and statistical techniques that are useful for the modeling and analysis of problems in which several variables have influence on the interested response. The main objective is to optimize that response.

Schwartz and colleagues [1970] demonstrate a technique for selecting the most desirable level of controllable variables in pharmaceutical tablet formulation so that the optimum properties are obtained. Their technique utilizes the statistical experiment design that is described by Box and Wilson [1987]. The data acquired from the experiment design is used to establish a mathematical model. This mathematical model, together with a feasibility search (i.e. graphical approach) is used to examine which formulation produces the optimal solution.

The five independent variables in this study consist of diluent ratio, compressional force, disintegration level, binder level, and lubricant level. Seven response variables include disintergration time, hardness, dissolution, friability, weight, thickness, porosity, and mean pore diameter. Note that these response variables are the desired tablet property. The experimental design employed here called "a five-factor, orthogonal, central, composite, second-order design" and the equation derived from running the experiment is a second-order polynomial.

Various authors have investigated the applications of the response surface method in pharmaceutical research and/or clinical pharmacology. Stewart, WH [1996] developed a combination of treatments to study the effective dose combination in hypertension using RSM and factorial design. Bota et.al [2000] studied the interaction effects of four variables in the adenosine deaminase enzyme assays and determined the optimal condition. Fassihi and co-workers [1995] examined the role of talc and magnesium stearate in the formulation of a controlled release system.

To this end, the proposed research's main focus is to optimize drug therapy. It is suggested that drug therapy can be optimized if a drug is given in the right dosage using the available information about the patients (i.e. state of health), the information about the drug dosage, and the treatments given to the patients. It can be seen that this work is closely related to that of response surface methodology. For instance, the problem can be viewed as the causal relationships between independent variables such as patient information, drug dosage, treatments and the response from the patient's symptoms. That is, there are several factors that have the influence in the response. Moreover, there are the interaction effects between the treatments and symptoms. In principle, once the relationship between different variables is known, then the optimal dose can be identified.

2.7 Response Surface Methodology

Response surface methodology (RSM) "*is a collection of statistical and mathematical techniques useful for developing, improving and optimizing processes*" [Myers and Montgomery, 2002]. The ultimate objective is to optimize (produce the maximum or minimum) values of the response. Response surface methodology is particularly appropriate in the situation where several input variables are likely to have an effect on the performance or quality of the product or process.

In any system, the conditions and settings that can be controlled during the process can be called *independent*, *factor* or *control* variables, normally represented by the symbol X. For instance, fertilizer and temperature could be considered as the control variables for growth in an agricultural experiment. Furthermore, the amount of fertilizer and degree of temperature are commonly referred to as *level*. Finally, the output or result of varying the control variables is known as a *dependent* variable, or *response*, denoted Y.

Typically, the shape of the response surface can be displayed graphically, either in three-dimensional space or as a contour plot. Each contour corresponds to a particular height of the response surface. Figures 12, 13, and 14 [from Khuri, Cornell, 1987] illustrate the surface plot in the three-dimensional space, the surface slicing, and the contour plot in studying the effects of the two fertilizers (X_1, X_2) on a peanut yield experiment (Y), consecutively. Note that these three figures are related. Finally, Figure 15 [Box and Draper, 1987] shows examples of different surface types of two variables (X_1, X_2) with second-order polynomials.

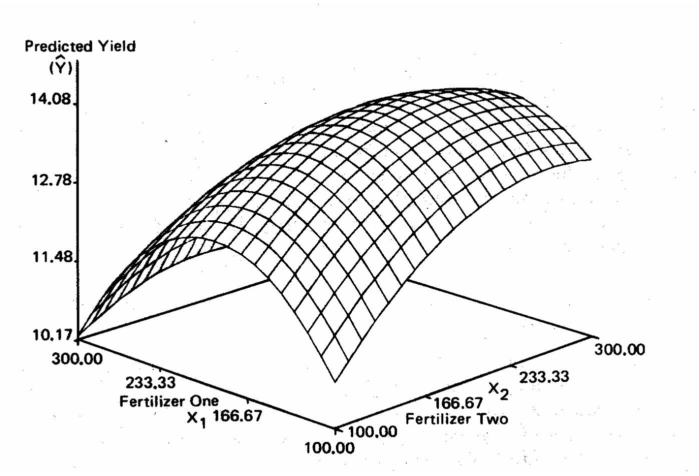


Figure 12: Three-dimensional plot of the peanut yield of two fertilizers (X1, X2) [Khuri and Cornell, 1987]

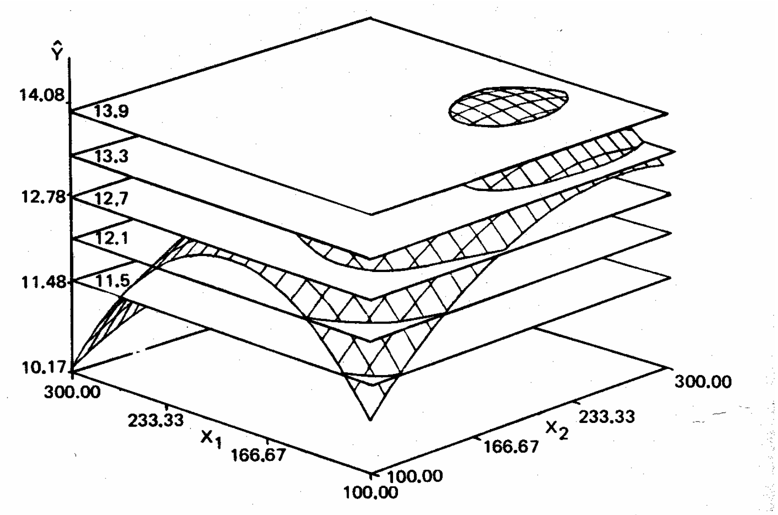


Figure 13: Slicing the surface at the different yields (11.5, 12.1, 12.7, 13.3, 13.9) [Khuri and Cornell, 1987]

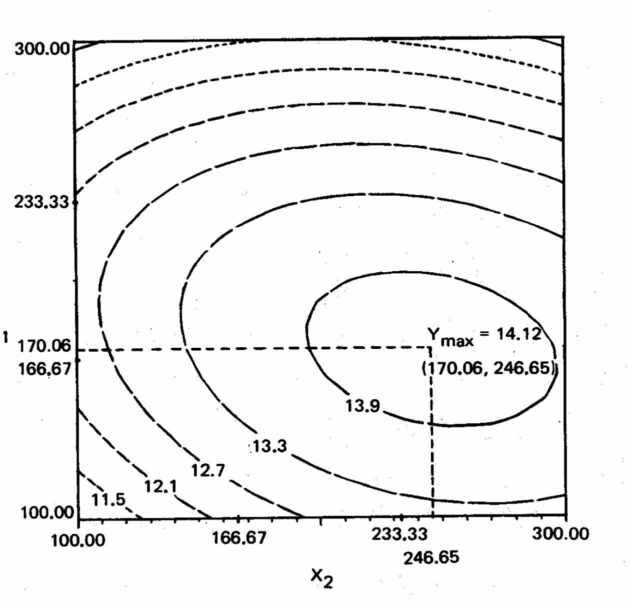


Figure 14: Contour plot of the yield surface [Khuri and Cornell, 1987]

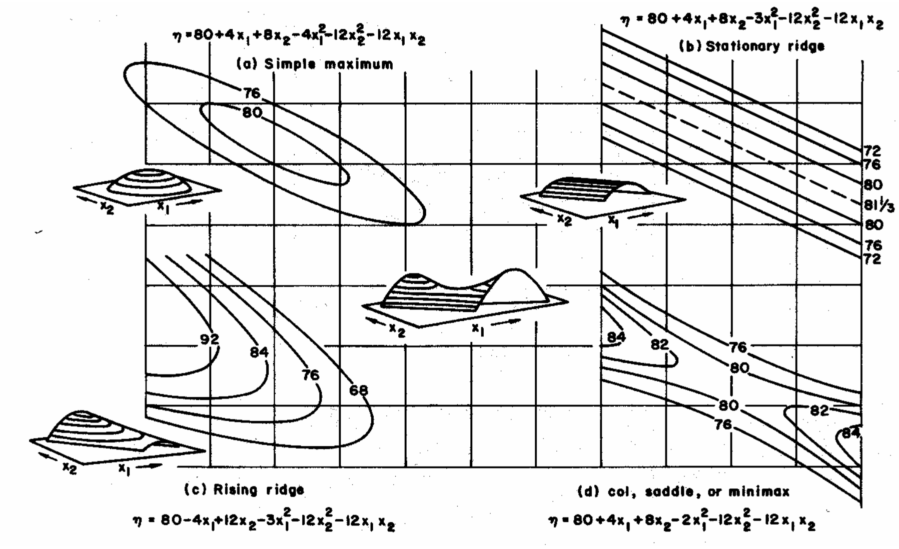


Figure 15: Examples of different surface types of two variables (X1, X2) with second-order polynomials [Box and Draper, 1987]

The work in response surface methodology can be traced back to the early 1930s. But it was not until 1951 that RSM was formally re-developed by a group at the Imperial Chemical Industries in England, led by G.E.P Box, K.B.Wilson, and others. Their aim was to investigate the relationship between the yield from the chemical process and input variables that were the main factors of that yield. Their work had a profound impact on numerous applications and was a motivator for additional research in fields such as engineering sciences, biological, clinical and social sciences. The comprehensive techniques used in RSM can be found in the book by Box and Draper [1987], and Myers and Montgomery [2002].

The RSM techniques for developing an empirical model can be summarized as follows: [Khuri, Cornell, 1987]

1) Design of experiments [Box and Draper, 1987] (usually abbreviated as DoE.) This is one of the most important tasks of RSM. The objective is to plan and conduct experiments such that the most amount of information can be extracted with the least number of experimental runs in order to identify the design variables that have the greatest influence on the interested response variables. As a matter of fact, the choice of the design of the experiments can have a great impact on the accuracy of the approximation model and the cost of construction the response surface [*http://www.brad.ac.uk/staff/vtoropov/burgeon*]. There are many types of experimental design, such as: full factorial design, fractional factorial design, and Plackett-Burman design. A detailed description of the type and design of experiments theory can be found in Box and Draper [1987], Myers and Montgomery [1995].

2) Establish a mathematical model that best fits the data derived from the design in step 1. This includes performing the appropriate hypotheses testing related to the parameters in that model. The resulting model can then be used to help understand the behavior of the process. General forms of the first order and second order polynomial can be presented as follows:

- First Order Approximation (Linear Model):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K$$

- Second Order Approximation (Quadratic Model):

$$Y = \beta_0 + \sum_{j=1}^{K} \beta_j X_j + \sum_{j=1}^{K} \beta_{jj} X_j{}^2 + \sum_{j=2}^{K} \sum_{i=1}^{j-1} \beta_{ij} X_i X_j$$

While these models show all of the possible terms, not all terms are necessary. It can be seen that, with higher degrees, the model becomes more complicated. Therefore, the model is usually limited to polynomials of either the first or second order.

3) Determine the experimental factors that generate the optimal (maximum or minimum) response value using a gradient search method.

The method of constructing the response surface is often an iterative process. After the approximation model is identified, its solutions will be tested for how well they fit. In the case of an unsatisfactory solution, the whole process is reinitialized and repeated. In fact, the experimenter's ability to develop a suitable approximation model that best fits the data is critical for the successful use of RSM [Myers and Montgomery, 2002].

Constrained mathematical optimization methods such as RSM may appear to be appropriate where all the pre-specified criteria have to be achieved simultaneously. However, there are times when simultaneous objects will be in conflict with each other. For instance, in designing a rocket component (in the field of aerospace engineering), the desired goals are increasing the performance, robustness, and safety, while attempting to decrease the overall weight and cost at the same time [Shyy, 1999]. RSM has been employed in a wide variety of applications; however, the research is focused mainly in medicine and its related fields.

This multi-objective optimization approach is widely used in pharmaceutical research, particularly during the drug design process. "Pharmaceutical product and process design problems are normally characterized by multiple objectives" [Fonners, 1970]. For instance, in the process of designing a drug product, there are many pre-specified objectives such as: unit cost, chemical stability, physical stability, and physiological availability of the active ingredients that could influence the dosage form characteristics.

To achieve all the pre-specified criteria simultaneously is a difficult task, especially when one response variable acts differently from another variable (i.e. one response tends to improve while another deteriorates). The desirable formulation for one property is not always desirable for the other characteristics. In other words, a conflict might arise when two or more objective functions are required to be optimized at once.

Various authors have investigated the application of RSM in pharmaceutical research and/or clinical pharmacology. Stewart, WH [1996] developed a combination of treatments to study effective dosages in treating hypertension using RSM and factorial design. Bota et.al [2000] studied the interactive effects of four variables in the adenosine deaminase enzyme assays and determined the optimal condition. Fassihi and co-workers [1995] examined the role of talc and magnesium stearate in the formulation of a controlled release system.

Generally, the true model of the system (the relationship between causal factors and individual pharmaceutical responses) is unknown. Therefore, the first step in system optimization is to determine the suitable approximation. To do so, the input variables that have the greatest impact on the response are identified. It is known that the fewer the number of variables that are involved, the easier it is to identify them. After the important variables are pinpointed, one can then find the best-fit model that represent the response as a function of these variables. Initially, low-order polynomials, such as the first order polynomial, are the most common form to be used. This initial model equation serves as the basis for later experiments. Knowledge from further experiments might suggest a different form of the equation, however.

As can be seen, the fitting and testing of models to determine the optimal solutions are repeated trial-and-error processes, and are often time-consuming. It can also costly because searching through all of the possible combinations of each individual variable in the model is often computationally expensive. Furthermore, the amount of data needed to evaluate the coefficients increase quickly with the order of the polynomial. The choice of the polynomial order and the terms to be used will depend on the design problem. A complete polynomial of *N* design variables with *m* order would require (*N*+*m*)! / (*N* !*m*!) coefficients. For example, a complete quadratic polynomial would require (*N+1*)(*N+2*)/(*2*!) coefficients [Del Vecchio, 1997]. There is even no guarantee that the correct model will be discovered. According to Takayama et.al [1999],"*prediction of pharmaceutical responses based on the second-order polynomial equation commonly used in RSM, is often limited to low levels, resulting in poor estimations of optimal formulations*".

The artificial neural network is a promising simulation tool that is able to detect the non-linear relationships between dependent and independent variables, which is the most common relationship in many applications, such as medicine and its related applications. An ANN works by learning the relationships among the variables in the data. In other words, the ANN implicitly learns how to relate the input variables to the output variables in a given problem. Additionally, the ANN is highly flexible in functional form and hence can offer significant potential for representing complex functions (not limited to the lower order polynomials as in RSM) as well as identifying all the possible interactions without complicated equations.

3. The Model

A number of researchers such as Stengel et.al.[2002], Barbolosi, et.al.[2001], Matveev et.al [2001], Iliadis et.al.[2000], Boldrini et.al.[1999] have used the optimal models to demonstrate the dynamic system of the disease. The proposed research focuses on the system control using the distal learning technique suggested by Jordan and Rumelhart [1992]. The optimal control structure utilized here is similar to that employed in many control systems, such as robotics and aircrafts. In this work, the technique of inversion contributes to the problem of finding the optimal value. Normally, the term 'inverse problem' or 'inversion' in control system refers to the problem of finding unknown quantity solutions based on effect values obtained from the observation. Notice that, this is called 'inverse kinematics' when used in the field of robotics.

Under the premise that the optimal treatments can be achieved if the right dosage of drug is given to a patient, the drug dosage is required to be optimized using the information available about: the patient's physiology, treatments (drug dosages), and the disease states (symptoms). Since treatments can have different effects on different patients, the adjustment of drug dosage according to an individual patient's need should result in maximizing the therapeutic efficacy and minimizing the side effects of the treatment.

3.1 Theoretical Approach

The distal learning technique is composed of two networks. Each network by itself is a feed-forward network using a back-propagation training algorithm. Typically, the first network is called *forward model*, whereas the second network is referred to as the *learner model*. Since this research is applied to medical applications, the names used for the models will be modified for a better understanding of the networks functions. Therefore, the first network will be called the *patient model (PM)* and the second network will be referred to the *treatment network (TN)* according to Munro and Sanguansintukul [2002]. The distal learning system can be illustrated as in Figure 16.

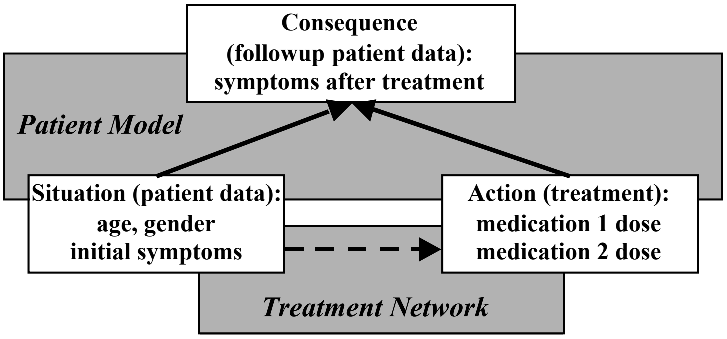


Figure 16: Distal learning system [Munro and Sanguansintukul, 2002]

In Figure 16, the feed-forward networks are represented with the shaded rectangles, while the patient data are in the white rectangles. The arrows indicate the information flow through the networks. The solid arrow lines indicate the inputs of the *PM*, while the dashed arrow denotes the input of the *TN*. Training the network can be divided into two stages: training the *PM* and training the *TN*. The details of training are discussed as follows:

First stage: training the patient model.

The purpose of this training is to predict the follow-up symptoms. The inputs to the network are the patient data (i.e. age, gender), symptoms (i.e. nausea, blood pressure), and the treatments such as doses from various medications. Given the information about the patient and the treatment, the network should be able to predict the follow-up symptoms in the patients. This is based on the assumption that there is a causal relationship between the treatment and its therapeutic effects (both beneficial and adverse effects). This relationship is also known as 'rational drug therapy'. It has been one of the main goals of clinical pharmacology to identify the optimum dose for a given disease state of an individual patient [Derendorf et.al.2000].

Training with a large sample of patient records allows the network to generalize its response to a larger population. This can potentially enhance the ability to identify effective treatment strategies. After the network has been trained to predict the follow-up symptoms, the weights in this network will be held fixed. This network will be integrated with the *TN* in the second stage.

Figure 18 displays the *PM* together with its input and output. It is important to note that all data (i.e. symptoms, treatments) must be converted into quantitative values before training.

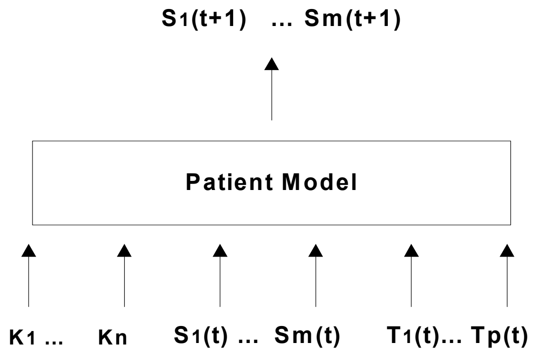


Figure 17: The patient model

In Figure 17, $\{K_1,\ldots,K_2\}$ represent a patient's personal data such as age and weight. $\{S_1(t), \ldots, S_m(t)\}$ indicate a patient's symptoms at time t, and $\{T_1(t),\ldots, T_p(t)\}$ denote a patient's treatments at time t. Finally, $\{S_1(t+1),\ldots, S_m(t+1)\}$ correspond to the symptoms at time $t+1$ (follow-up symptoms).

The goal of training is the network prediction performance: what the symptoms of a patient will be at time $t+1$, if the patients' personal data, symptoms and treatments at time step t are given. It is fair to assume that doctors recommend good treatments to patients, since they are experts in the field. The expectation of *PM* training is that the prediction performance should be at least equivalent to experts. After the *PM* is trained, the connection weights will be frozen and employed in the next step.

Second stage: training the treatment network.

The objective of training the *TN* in this stage is to optimize the follow-up symptoms. In this case, the focus is on minimizing the symptoms. In order to achieve this requirement, the *TN* has to learn how to adjust the treatments so that in the end the symptoms are minimized. This is under the assumption, mentioned earlier, that the therapeutic effects are the result of efficacy of the treatments. In other words, the optimal dosage will cause the symptoms to be minimized. The treatment network is illustrated in Figure 18.

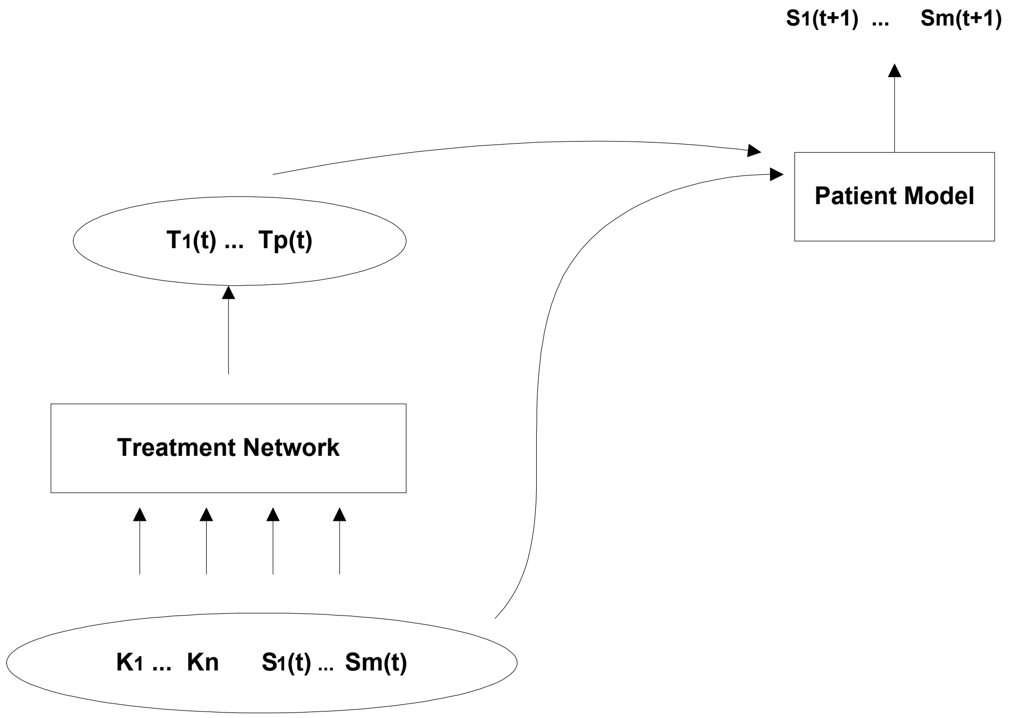


Figure 18: Treatment network

To be specific, the *TN* is trained and its connection weights are adjusted to optimize the *PM* performance or generate the ideal training signal (optimal treatments) to the *PM*. Implementation detail consists of two-iteration steps.

- First step

TN performs treatment prediction. The training inputs are patient's data and symptoms $\{K_1,\ldots,K_n;\ S_1(t),\ldots,S_m(t)\}$ and the outputs are the predicted treatments $\{T_1(t),\ldots,\ T_p(t)\}$ from the *TN*. These predicted results will be used as part of the inputs in the next step. The relationship between inputs and outputs of the *TN* can be written as:

$\{K_1,\ldots,\ K_n;\ S_1(t),\ldots,\ S_m(t)\} \rightarrow \{T_1(t),\ T_p(t)\}$

- Second step

Error is back-propagated from *PM* to *TN* and weights in the *TN* are adjusted. To measure the performance of the *PM*, the patient's data, symptoms and treatment predicted from the previous step are given to the *PM*. If *TN* generates the ideal training signal, the follow-up symptoms from the *PM* should be minimized. Note that the weights in the *PM* are kept frozen. Only weights in the *TN* are allowed to be changed.

The relationship between inputs and outputs of the *PM* is:

$\{K_1,\ldots, K_n; S_1(t),\ldots, S_m(t); T_1(t), T_p(t)\} \rightarrow \{S_1(t+1),\ldots, S_m(t+1)\}$

Even though the weights in the *PM* will not be adjusted, the errors are still back propagated to the *TN*. Munro [2002] explained that when the weights of the *PM* are fixed, the *TN* is forced to figure out the appropriate dosages that produce the optimize symptoms. Otherwise, the network could arrive at the optimal symptoms solution by simply ignoring the input and adjusting only the output biases.

The employment of this optimization technique in medical applications has been proposed. This technique is inspired by the inverse kinematics used in the robotic control system. The goal is to identify the optimal treatment, which could minimize the likelihood of the unfavorable response; simultaneously, it is possible to maximize the favorable state of health in the patients. Due to the fact that treatments can have different effects on different patients, utilizing all the available information of individual patients in the training of the networks could contribute to success in prescribing the treatments. Finally, this methodology may offer a way to blend the drug effects, while minimizing the cost of the treatment, as well as maximize the physiological health of the patients.

4. Methods and Results

In medicine, the choice of treatment is critical. The studies described here evaluate the feasibility of a specific kind of artificial neural network architecture as an assistive aid in determining treatment.

The experiments include three studies, all based on the architecture described in Section 3.1: the theoretical approach (Chapter 3). Study 1 employs the network for optimization problems with one objective function (with and without interaction). Study 2 examines multi-objective optimization problems that have one treatment (with no interaction) and an analysis of coefficients in the PM and TN. Study 3 is the study of multi-objective functions that have multiple treatments and interactions between those treatments.

A mathematical model was created in order to simulate patient states and treatments during the first cycle of treatment. The networks are trained using data generated as a function of this model. When given a schedule of drug administration, the network should be able to optimize the treatment by limiting the drug toxicity that could worsen a patient's symptoms. In other words, the network should be able to find the optimal dose(s) that minimize all symptoms in the patient. The variables used in all studies are summarized in the following table:

Variable	Symbol	Computation
Age	‘a’	Normal distribution with a mean of 50 and SD of 8.
Gender	‘g’	Randomly generated with equal probability of 0.5 for male or female.
Initial Symptom	‘s_i’	Uniform distribution in the range of [0,1], unless otherwise specified.
Follow-up Symptom	‘s_f’	See equations [7], [10], and [11]. (Studies 1-2 only)
Follow-up Result	‘r_i ‘	See equations [15], [16], and [17]. (Study 3 only)
Treatment	‘d_i’	PM: Uniform distribution in the range of [0,1]. TN: The predicted dose(s) from the TN network.

In studies 1-2, it is important to note that the PM explicitly predicts follow-up symptom(s) s_f from given initial symptom(s) s_i. In study 3, The PM predicts r_i, the ratio of s_f to s_i without any input representation of s_i.

4.1 Study 1a (Preliminary Study): One objective function using two doses with no interaction

4.1.1 Methods

Each item (α) of data consists of six variables: age (a), gender (g), symptom (s_i), dose 1 (d_1), dose 2 (d_2) and the follow up symptom (s_f). The first three variables represent the initial state of the patient, the next two variables denote the two drug treatments, and the last variable is the resulting follow-up symptom. The follow-up symptom (s_f) is computed from equation [7].

$$s_f = s_i * (1 - \exp - \left[\frac{\left(d_1 - d_1^{opt}\right)^2}{k_1} + \frac{\left(d_2 - d_2^{opt}\right)^2}{k_2} + \rho(\varepsilon) \right]) \qquad [7]$$

Where k_1 and k_2 are constant (here both are equal to 1/10).

The equations [7], [8], and [9] to which this section refers come from Munro and Sanguansintukul[2002]. The optimal dose for treatment 1 (d_1^{opt}) is assumed to be a function of age. Here, the optimal dose $d_1^{opt} = age/100$. On the other hand, the optimal dose for treatment 2 (d_2^{opt}) is assumed to be a function of gender. Here, the optimal dose 2 for a male is $(d_2^{opt})=0.7,$

while the optimal dose 2 for a female is $(d_2^{opt})=0.3$. The symbol $\rho(\varepsilon)$ denotes noise, which can be added to the target value (s_f) and is drawn uniformly from the interval $[-\varepsilon, \varepsilon]$.

The data set is comprised of 6000 patient records, of which 5000 records (N_{Tr}) are used as the training set and the remaining 1000 records (N_{Ts}) are used as the testing set. All variables in the data set are scaled to the range [0,1]. The purpose of the training is to optimize the follow-up symptom (s_f). It is possible to extend the function to include multiple symptoms and treatments. In this study, each treatment can have its own optimal value. The training of the networks is divided into two stages: PM and TN.

First stage: PM training

In the patient model, the focus is on training the network so that it can predict the follow-up symptoms. The patient model is a feed-forward network that consists of 5 inputs (a_1, g_1, s_1, d_1, d_2) and 1 output (s_f), trained using the back-propagation learning algorithm. The objective function intended to be minimized by the network is shown in equation [8].

$$\sum_{\alpha}^{N_{Tr}} \left(s_f^{\alpha} - r_{PM}\left(a_i^{\alpha} g_i^{\alpha} s_i^{\alpha}, d_1^{\alpha}, d_2^{\alpha}\right)\right)^2, \qquad [8]$$

where r_{PM} is the output of the PM.

The network is provided with 5000 records for training. After the network has been trained, the resulting weights are frozen. The network performance is then tested using another 1000 records. The patient network is illustrated in Figure 19.

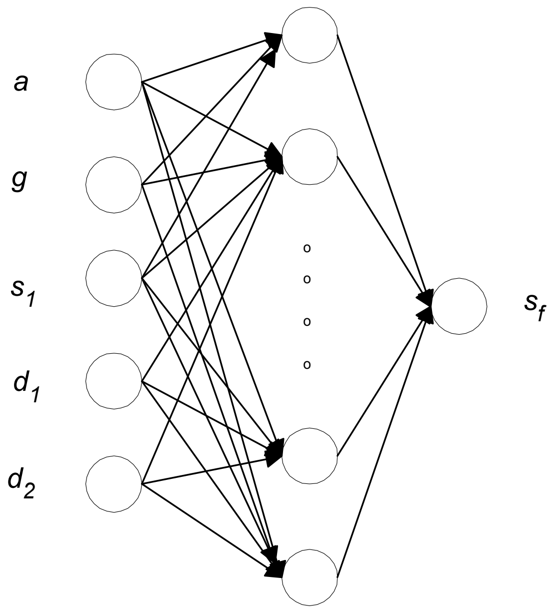


Figure 19: Patient model

Second stage: TN training

The primary network is called the *treatment network*. The purpose of this network is to predict the treatments (doses) that could optimize the output of the patient model (the follow-up symptom). The treatment network is also a feed-forward network that employs the back-propagation technique. The learner portion of the treatment network consists of 3 input units (*a_1, g_1, s_1*) and two output units (*d_1, d_2*). As mentioned earlier, the learner network aims to optimize the output of the patient model. The objective function used is shown in Equation [9].

$$\sum_{\alpha}^{N_{Tr}} \left(s^* - r_{PM}\left(a_i^{\alpha}, g_i^{\alpha}, s_i^{\alpha}, r_{TN}\left(d_1^{\alpha}, d_2^{\alpha}\right)\right)\right)^2, \qquad [9]$$
$$\text{where } s^* \text{ is the desired symptom value, and } r_{TN}$$
$$\text{is the output of the } TN \text{(dosage values).}$$

Figure 20 shows the integrated network employed in this stage. The treatment network is placed in series with the patient model, represented by solid and dashed lines, respectively. The patient network has been previously trained to predict the follow-up symptom when given patients' states and treatments. This composite learning system can be considered a single supervised learning system. The patient network should not be allowed to change the weights while the composite learning system is being trained so that the learner can find the optimal treatments (doses) by varying only the mapping of patient states to treatments.

The network with the solid line connections is the treatment network. This network consists of 3 input units (a_1, g_1, s_1) and two output units (d_1, d_2). The network with the dash-line connections is the patient model that was trained from the previous stage but now has fixed weights. The 5 inputs to the forward network are (a_1, g_1, s_1, d_1, d_2). The first three are the same inputs used in the first stage. However, the last two inputs (d_1, d_2) are the outputs from training the treatment network.

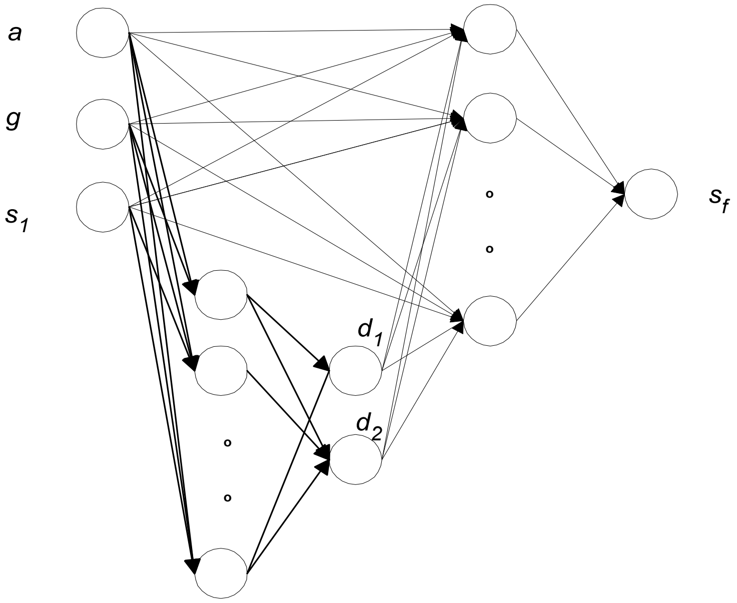


Figure 20: Treatment network

4.1.2 Results

Patient Network

Analysis was performed to determine the optimal values for three parameters associated with back-propagation training: learning rate, momentum and the number of hidden units. It was determined that the optimal values of these parameters, for the both the patient model and the learner network, are 0.01, 0.8, and 6. From the experiments, the initial weights should be in the range [-0.001, 0.001]. No noise is used in this study ($\varepsilon = 0$).

There are two conditions under which the training will be stopped: the network trains for 5000 epochs, or the sum square error goes lower than 0.0001. Here, the network is trained for 5000 epochs and was then stopped. The weights were then fixed and the network was presented with the testing set to examine the generalization performance.

Figure 21 illustrates the relationship between the predicted outputs (follow-up symptoms) from the network and the corresponding targets. There are altogether 1000 points on the graph. The ideal points are expected to lie along *x*=*y*, which is the dashed line on the diagonal of the graph. The linear fitting line for the points on the graph is shown as a solid line. It can be seen from the graph that the fitting line is very close to ideal. This indicates that the network can perform generalizations even with previously unseen data.

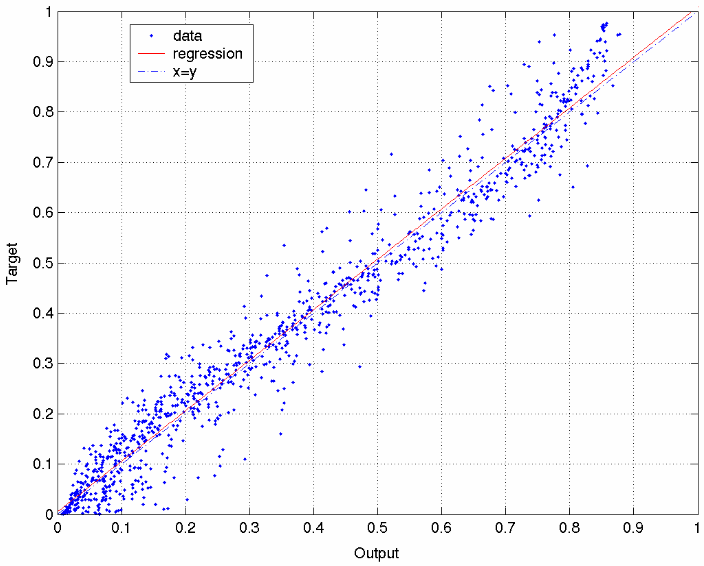


Figure 21: Target values and predicted value of the patient network.

Figure 22 shows contour plots for a 25-year old female. The solid contour line is the desired function, while the dashed contour line is the predicted function from the network. The asterisk ‘*’ indicates the ideal value, while the plus ‘+’ represents the predicted value of the network.

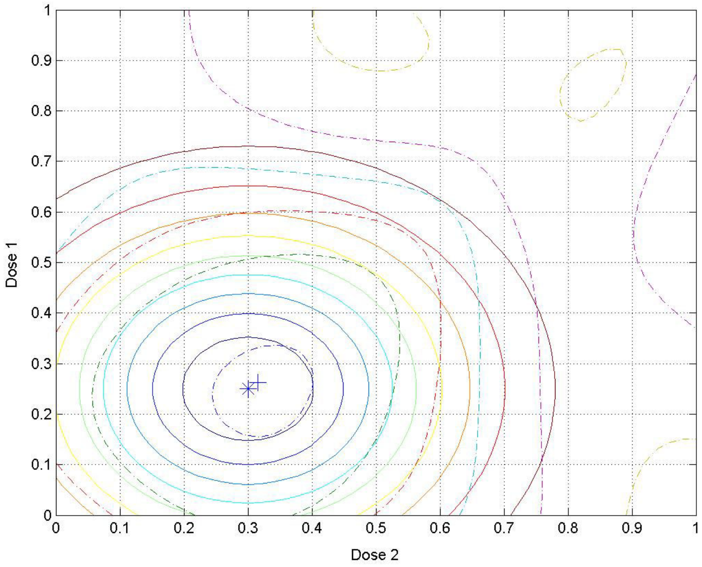


Figure 22: Contour plots of the target function and predicted function for a 25-year old female

Treatment Network

To evaluate the prediction performance in the treatment network for symptom optimization, the predicted output 1 (dose 1) and output 2 (dose 2) and their corresponding targets are plotted.

Figure 23 demonstrates the relationship between output 1, the predicted values, and target 1. The ideal point or optimal value for target 1 was specified in equation [7], which is equal to age/100. Once again, the ideal line *x*=*y* (dash line) and the fitting line (solid line) are both displayed in the graph. It is seen that the fitting line is very close to ideal.

Figure 24 displays the relationship between output 2 and target 2. The ideal point or the optimal value for target 2 is 0.7 for males and 0.3 for females. Again, the predicted output is close to the ideal. The close correspondence of these two cases implies that the treatment network is capable of computing the optimal dose.

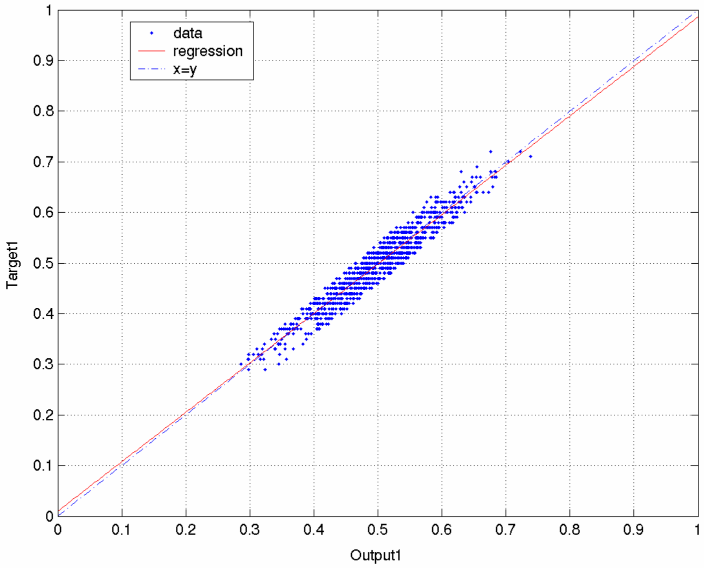


Figure 23: Target 1 and output 1 (predicted dose 1) of the treatment network.

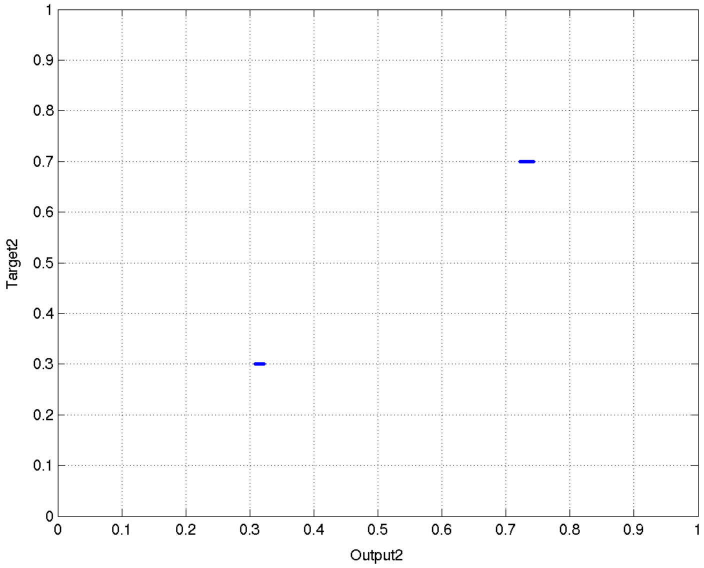


Figure 24: Target 2 and output 2 (predicted dose 2) of the treatment network.

To further evaluate the performance of the treatment network, the ideal dose for dose 1 and dose 2 predicted by the network and the ideal values are compared in Figure 25. Six combinations from two genders (male, female) and three age levels (25, 45, 65 years old) are displayed in the graph. The contour plot associated with each case is omitted. Only the ideal and predicted doses are plotted. The ideal doses are signified by the symbol '*', while the predicted doses from the network are denoted by symbol '+'.

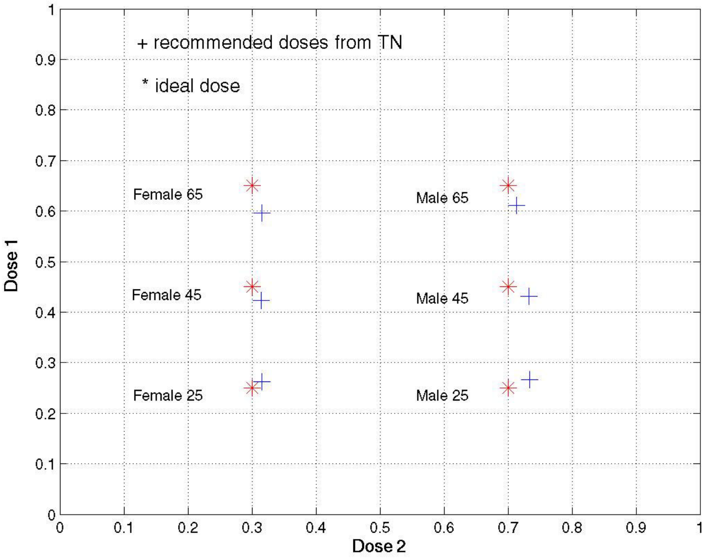


Figure 25: Computed doses from the treatment network compared with ideal doses for six age-gender combinations.

4.2 Study 1b: one objective function using two doses with interaction

This is an extension of the preliminary study, in which neural networks have been employed in optimization problems with one objective function and two different treatments where those treatments have an interaction with each other.

4.2.1 Method

Each item (α) of data consists of six variables: age (a), gender (g), symptom (s_1), dose 1 (d_1), dose 2 (d_2), and the follow up symptom (s_f). The first three variables represent the initial state of the patient, the next two variables denote the two drug treatments, and the last variable is the resulting follow-up symptom. The follow-up symptom (s_f) is a modification of equation [7] in the preliminary study with the addition of an interaction. It is illustrated in equation [10] as follows:

$$s_f = s_i * [(1 - \exp - [\frac{((d_1 + d_2) - D_1)^2}{k_1} + \frac{((d_1 - d_2) - D_2\ )^2}{k_2} + \rho(\varepsilon)])] \qquad [10]$$

Where k_1 and k_2 are constants.

When $D_1 = d_1^{opt} + d_2^{opt}$ and $D_2 = d_1^{opt} - d_2^{opt}$, the function in equation [10] is equal to zero. For the purpose of this study and to ensure that the dose values are not negative, the age will be encoded in the range [0.4, 1.6]. In this study, the optimal values for male and female are 0.4 and 0.2, respectively. When solving the above equation, d_1 is in the range [0.3, 1.0] and d_2 is in the range [0.0, 0.7]. Noise $\rho(\varepsilon)$ is drawn uniformly from the interval [-ε, ε] when ε is between [0, 1].

4.2.2 Results

Figures 26-28 illustrate the performance of the PM and the TN networks after training.

PM performance

The PM was trained with 7000 epochs, a learning rate of 0.05, a momentum of 0.9, while using 7 hidden units and a linear activation from hidden to output units. After training, the performance of the PM is evaluated by plotting the output (predicted follow-up symptom) against the corresponding target value on a testing set as shown in Figure 26.

Ideally, if the output values exactly match the desired target values, the points should lie along the diagonal line. From Figure 26, the diagonal line (when x = y) is shown in solid and is considered as an ideal line. The dashed-line in the graph, also called the regression line, shows the expected output values. In terms of the average response, the network performs quite well, as shown from the regression line being very close to the ideal line.

Figure 27 shows a contour plot as a function of two treatments for a 45 year-old male. The solid curves are contours of target functions, while the dashed curves are contour functions computed by the PM. These contour curves are very close to each other, particularly near the optimal point. It should be noted that the shapes of the contours in Figure 27 are different from the shapes of contours in the preliminary study (Figure 22) due to the interaction effects between treatment doses.

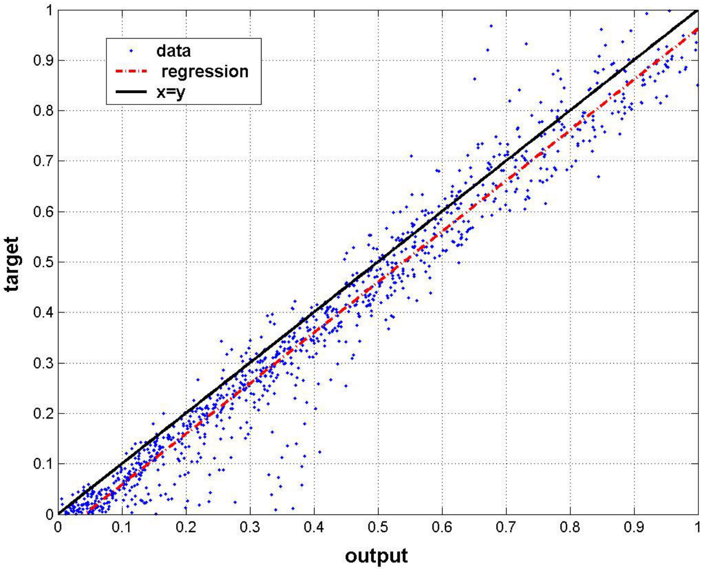


Figure 26: Plot of predicted and ideal symptoms of the PM.

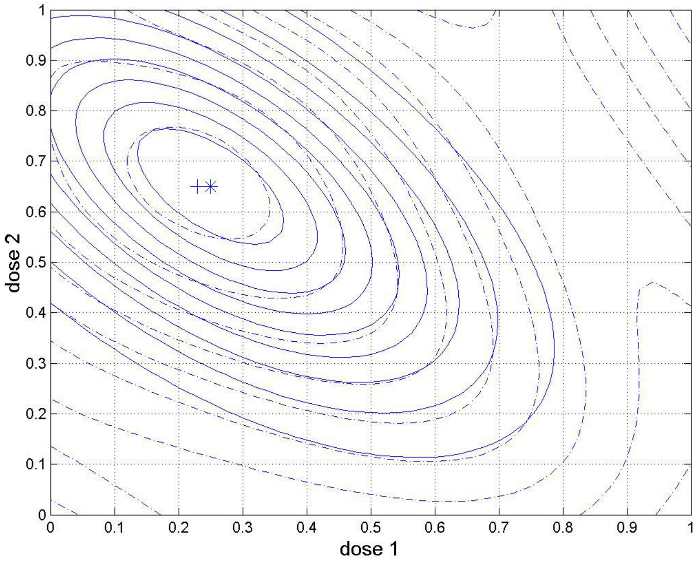


Figure 27: Contour plot of a 45 year old male.

TN performance

The TN performance is evaluated by plotting doses recommended by the TN against the ideal doses, as displayed in Figure 28. The symbol '+' signifies doses predicted from the network, while the symbol '*' represents ideal doses calculated from a function. The graph shows the six combinations of gender (M/F) and age (25, 45 and 65 years old). The results from the network are very impressive, as the predicted doses are very close to the ideal points. Here, the distance between ideal and predicted doses is used as a performance measurement of the TN and the whole system.

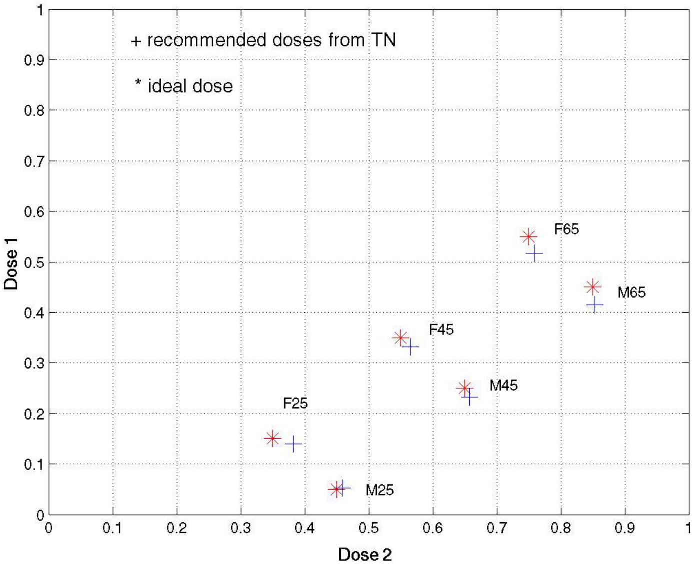


Figure 28: Predicted doses from the TN.

The effects of noise and sample size were chosen for investigation in study1b because the analysis can be comprehended better when there is one symptom involved. However, it is not too simple since there is an interaction between combined treatments.

4.2.3 Effect of sample sizes

4.2.3.1 Methods

The performance of the PM on different sample sizes was investigated. The sample sizes ranged from small to large, consisting of 100, 300, 500, 1000 and 3000 records that were chosen for evaluation purposes. For each sample size case, ten different data files were generated. All of these data files, a total of 50, were used to train the PMs. Note that the equation used for creating these data files is based on equation [10]. After training the PMs, the weight files corresponding to each data set were saved for later use.

In addition to these 50 different files, another data file consisting of 20,000 records was generated as a test set. Each record consisted of a group of five factors: age, gender, symptom 1, dose 1, and dose 2. The number of records is derived from the combination of ten different values of age, two different values of gender, and ten different values each for symptom 1, dose 1, and dose 2. This test data, along with each saved weight file from the previous process, were used to produce an output file. The average difference between the target and output values of the 10 different data files for each sample size case was calculated.

4.2.3.2 Results

The results can be seen in Figure 29, which depicts the sample size and its corresponding error. The x-axis represents the sample size, while the y-axis shows the error (deviation from the target values). For each sample size, the error of each simulation is denoted with symbol ‘x’. The average error of each sample size is illustrated with a point where the (horizontal) line is connected between sample sizes.

It can be seen that the sample size of 100 has the highest average error, while the sample size of 3000 has the lowest average error. Generally, it can be seen that the smaller the sample size, the higher the average error. However, the average error of sample sizes 300 and 500 differ only slightly. This is also the case for sample sizes 500 and 1000. In addition, the outlier occurs when small sample size of 100 and 300 were employed.

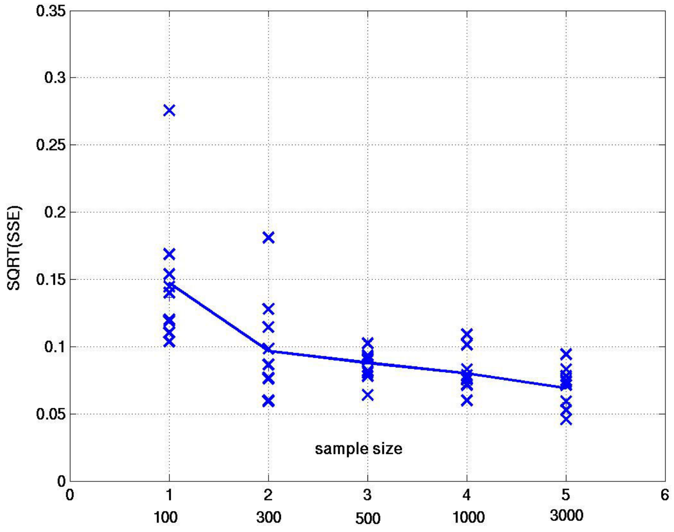


Figure 29: Average error of different sample sizes (no noise)

4.2.4 Effects of sample sizes and noise

4.2.4.1 Methods

To further explore the performance of the PM, noise in the range of [-0.1, 0.1] was introduced. Data files of sample size 100, 300, 500, 1000, and 3000 records from the previous section were employed, but with noise added to the data. Each data sample size was composed of another 10 different added noise files. The PMs were trained with these data files and the weight file of each training simulation was saved. The test data file used in the prior experiment with no noise was used again in this study. The average difference between the target and output values of 10 different data files with noise for each sample size case was determined.

4.2.4.2 Results

Figure 30 illustrates the sample size with noise and its corresponding error. Different sample sizes are displayed on the x-axis, while the error is shown on the y-axis. The error from each simulation of each sample size is displayed with the symbol '+'. The average error of each sample size is illustrated with a point where the (horizontal) line is connected between sample sizes.

Analogous to the case without noise, the smaller sample size with noise tends to have a higher average error. Among the sample sizes used in this study, size 100 again shows the highest average error. The average error of sample sizes 300 and 500 is comparable. This is also true for the case where sample size is 1000 and 3000. Additionally, the smaller sample size appears to have a higher range variation of error than the larger sample size.

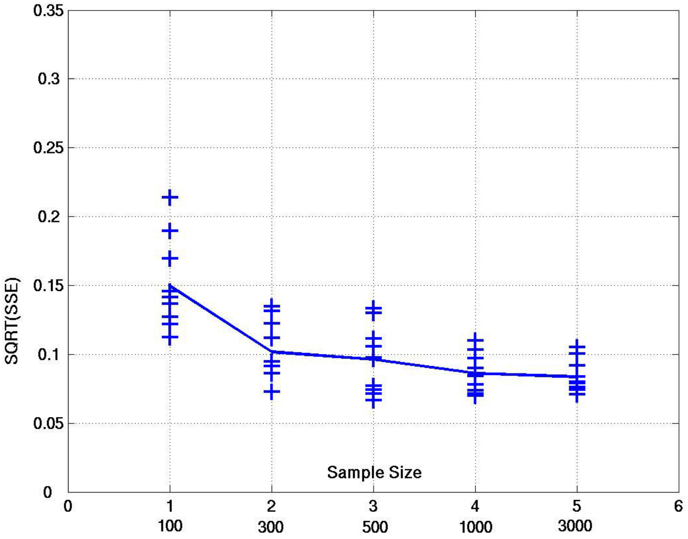


Figure 30: Average error of different sample sizes (with noise)

The comparison between the average errors with noise and those without noise is demonstrated in Figure 31. The '*' symbol signifies the case where there is no noise. Correspondingly, the triangle denotes the case where noise is present. The graph indicates that the average error of sample sizes with no noise is slightly lower than when there is noise. The tendency for error is lower when the sample size is larger in both cases (with and without noise).

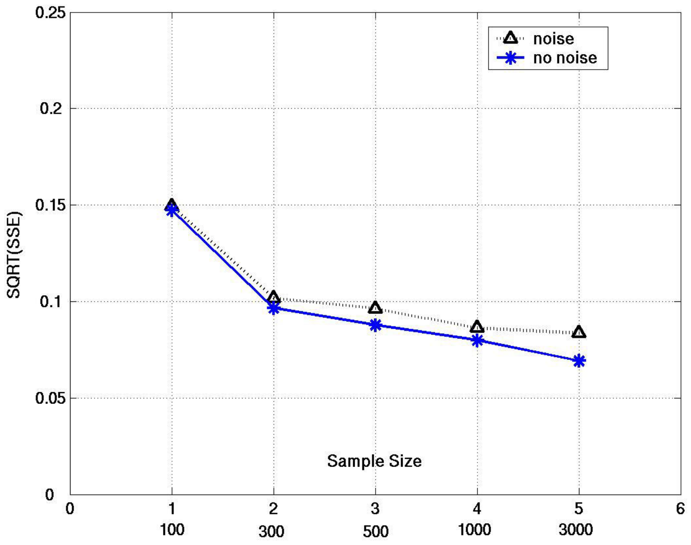


Figure 31: Compare average error between noise and no noise

Nevertheless, the range of error where noise is present tends to have a lower variation than where there is no noise (accounting for the outlier for sample sizes 300 and 500 when there is no noise present). However, it is more obvious when the larger sample sizes of 1000 and 3000 were applied. The comparison of error range for noise and no noise can be seen in Figure 32.

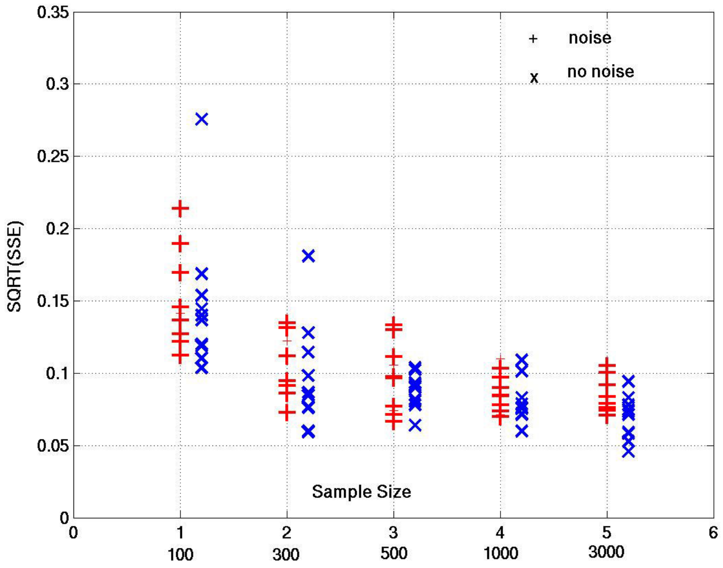


Figure 32: Compare range of error for noise and no noise

4.3 Study 2: multi-objective functions using one dose with no interaction

This study addresses multi-objective functions in treatment optimization. Achieving multi-objective criteria goals is difficult because solutions are based upon finding the best possible combinations of control variables (such as dose) that optimize different objectives simultaneously. This means that the solution is often a compromise between possibly conflicting objective requirements.

Some medical applications that deal with multi-objective criteria are cancer treatments, such as chemotherapy and radiotherapy. In cancer chemotherapy, the central concern is to design drug regimen treatment strategies that ensure that tumor cells are killed at a desired rate. At the same time, the dose given should not cause an unacceptable toxicity to the host (patient). Since chemotherapy simultaneously kills tumor cells and causes toxicity in normal cells, optimal treatment can be achieved if a drug is given in the right dose that maximizes the former while keeping the latter within acceptable limits. In the case of external radiotherapy, delivered radiation passes through normal as well as tumor cells. Therefore, an ideal goal is to deliver a dose that kills all tumor cells while avoiding damage to surrounding healthy issues.

For the purpose of this study, a mathematical model was created in order to simulate patient states and treatments during the first cycle of treatment. Radiotherapy treatment outcome indices, namely, tumor control probability (TCP) and normal tissue complication probability (NTCP), are utilized as the objective functions.

The radiobiological response of the tumor is usually depicted as a sigmoidal relationship between delivered doses and the TCP (see Figure 33). In Figure 33, the TCP increases as the delivered dose increases. As the tumor cells are killed, the delivered doses of radiation also cause damage to the surrounding healthy tissues. Figure 34 shows the relationship between delivered doses and the NTCP, which is also sigmoidal. In Figure 34, three different points (A, B, and C) denote TCP-10, NTCP-10, and TCP-90 respectively. Presumably, point B is the maximum for the acceptable normal tissue complication rate. Therefore, the TCP is limited by point B. The critical problem is finding the maximum delivered dose that kills tumor cells as well as minimizing damage to the surrounding tissues. Details about dose optimization in radiotherapy can be found in http://www.varian.com/onc/prd061_3.html.

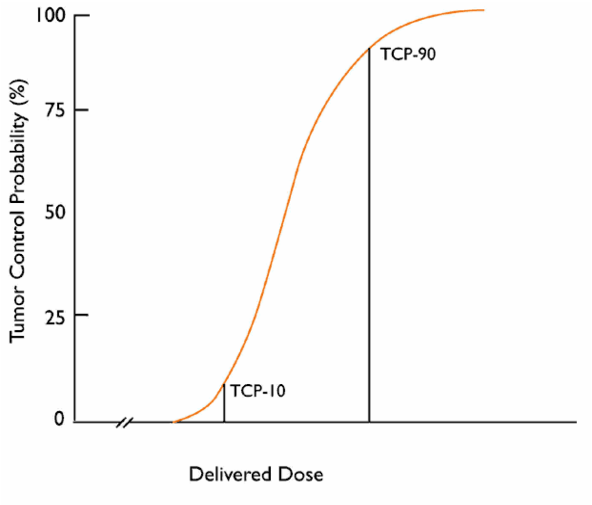


Figure 33: Delivered dose and Tumor Control Probability (from: http://www.varian.com/onc/prd061_3.html)

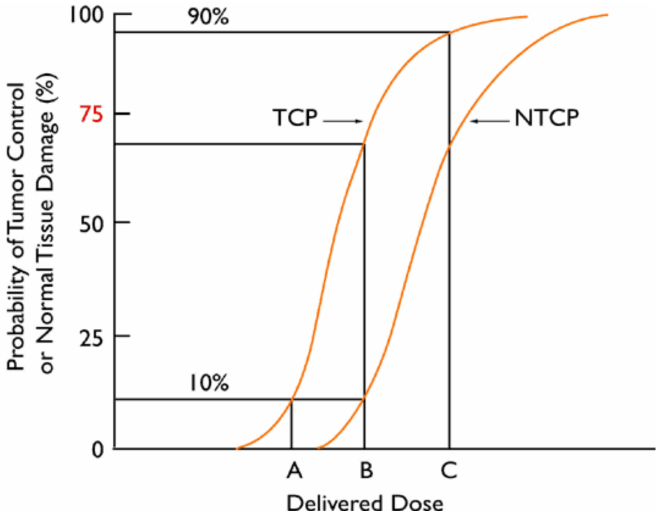


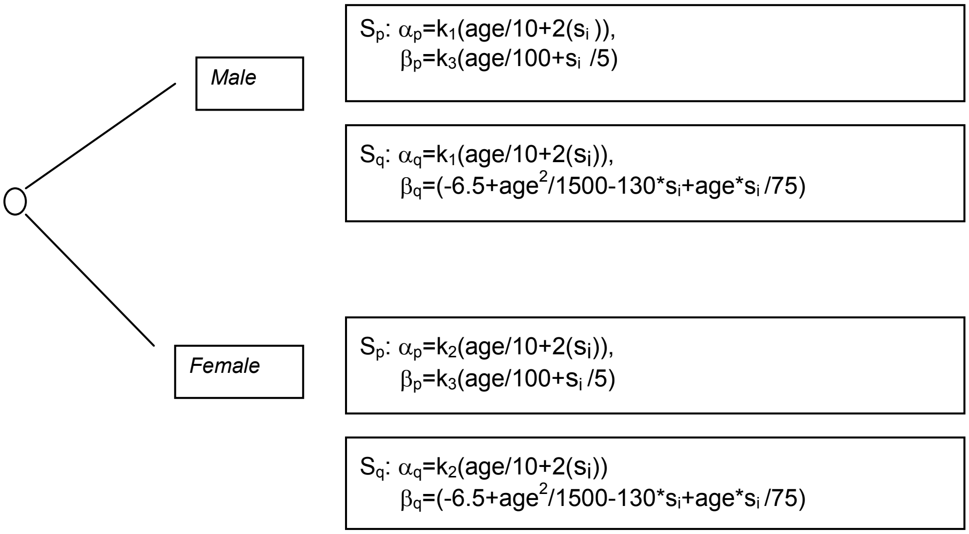
Figure 34: Delivered dose and NTCP (from: http://www.varian.com/onc/prd061_3.html)

Generating artificial data for two objective functions

Each record (α) consists of six variables: age(a), gender(g), tumor size(s_i), dose(d), follow-up symptom 1 or TCP (s_p), and follow-up symptom 2 or NTCP (s_q). The first three variables represent the initial state of the patient, while the next variable denotes the treatment, and the last two variables are the resulting follow-up symptoms. The tumor size was generated from a normal distribution with a mean of 0.6 and a standard deviation of 0.1. The follow-up symptoms (s_p, s_q) are computed using the following function:

$$s_x = \frac{1}{1+e^{(-\alpha_x * d - \beta_x + \rho(\varepsilon))}} \qquad where\, x = \{p,q\} \qquad [11]$$

In this situation, each symptom x has a functional dependence on a dose that is a logistic function with noise (ε) added to account for individual variability. (ε) is drawn uniformly from the interval [-ε, ε] where ε is between [0,1]. The values of α_x and β_x depend on age, gender, tumor size, and symptom. In addition, the values vary according to following conditions:



Where k_1, k_2, k_3 are constants

Figure 35: conditions for generating following symptoms (s_p, s_q)

The functions are formulated under the assumption that the larger the tumor size, the more delivered dose is needed. In addition, younger patients tend to tolerate higher doses than those who are older. Finally, those who are male tend to better endure higher doses.

The graphical plots of example cases are illustrated in Figures 36 and 37. Figure 36 shows two cases where the patients are 50-year-old males having tumor sizes of 0.2 and 0.8, respectively. The solid lines on the left and right represent S_p and S_q, respectively, for the patient with tumor size 0.2. The dash-lines on the left and right are S_p and S_q for the patient with tumor size 0.8.

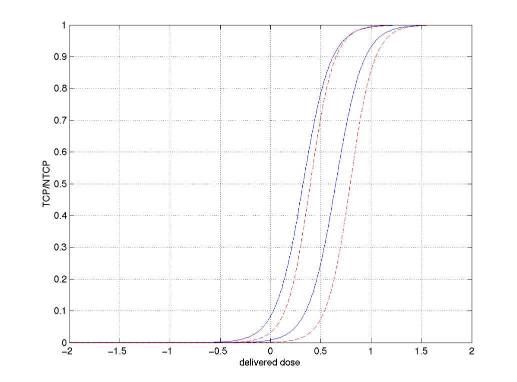


Figure 36: Sp Sq for male 50yrs, tz=0.2,0.8

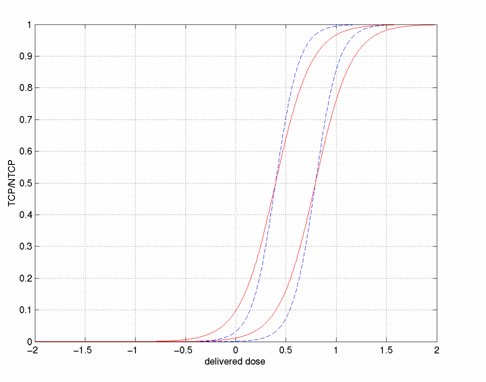


Figure 37: Sp Sq male, female 50 yrs, tz=0.8

In Figure 37, the solid lines on the left and right are S_p and S_q generated from a case where the patient is a 50-year old female with a tumor size of 0.8. The dashed-lines on the left and right represent S_p and S_q from a case where patient is a 50-year old male with a tumor size of 0.8.

The objective function that is to be minimized for the patient network is shown in Equation [12].

$$\sum_{i}^{2}\sum_{\alpha}^{N_{tr}} A_i(s_{f_i} - r_{PM_i}(a^{\alpha}, g^{\alpha}, tz^{\alpha}, d^{\alpha}))^2 \qquad [12]$$

Here, N_{tr} is the number of training records; s_{f_i}, when i is {1,2}, denotes follow-up symptoms (s_p, s_q), respectively; r_{PM_i} is the output of the patient network; and finally, A_1 and A_2 are the relative important factors (or weight factors), where $A_1 + A_2 = 1$, that are applied to the patient network to emphasize how precise the symptoms need to be.

Equation [13] shows the objective function to be optimized for the treatment network.

$$\sum_{i}^{2} \sum_{\alpha}^{N_{tr}} B_{\,i}(s_i * - r_{PM_{\,i}}(a^{\alpha}, g^{\alpha}, tz^{\alpha}, r_{TN_i}(d_1{}^{\alpha})))^2 \qquad [13]$$

Here, s_i^*, when i is {1,2}, is the desired symptom value; r_{TN} is the output of the treatment network; and finally, B_1 and B_2 are the relative importance factors (or weight factors), where $B_1 + B_2 = 1$, that are applied to the treatment network to optimize the objective function.

4.3.1 Study 2a: Analysis of coefficients (A_1, A_2) in the patient network

4.3.1.1 Methods

Different coefficient pairs are used to the train the patient network. The coefficient pairs (A_1, A_2) used are: (0.1, 0.9), (0.2, 0.8), …, (0.9, 0.1). This analysis can be used in the situation where a clinician wants to emphasize the prediction accuracy of one symptom over the other.

4.3.1.2 Results

This experiment examines the effects of treatment dose on multiple symptoms. The PM consists of two output units, each unit corresponding to a symptom (TCP and NTCP). Each symptom can be treated with different weights. For example, a coefficient pair of (0.1, 0.9) means that the first symptom (TCP) is assigned with the first coefficient weight of 0.1, while the second symptom (NTCP) is assigned with the second coefficient weight of 0.9. The network is trained using different coefficient pairs: (0.1,0.9), (0.2,0.8), …, (0.9, 0.1). In the process of training, these coefficient pairs were applied to the objective (cost) function of the network.

In order to show the effect of different coefficients on the cost function of the PM network, the sum square difference between the PM outputs and their desired targets, for each coefficient pair, is computed and compared, as shown in Figure 38. From Figure 38, different coefficient pairs of symptom 1 (white bar) and symptom 2 (dark bar) are displayed on the x-axis, while the y-axis represents the sum square error. For example, the two bars on the far left represent the sum square error of symptom 1 and symptom 2 when a coefficient pair (0.1, 0.9) is applied to the cost function. From the graph, the SSE of symptom 1 using (0.1, 0.9) as a coefficient is higher than the SSE of the same symptom using (0.2, 0.8). Similarly, the SSE of symptom 1 using (0.2, 0.8) is higher than the SSE of symptom 1 using (0.3, 0.7), and so on. In the same manner, symptom 2, which is trained with (0.9, 0.1), has a higher SSE than the same symptom trained with (0.8, 0.2). In general, the sum square error of both symptoms tends to be lower when the higher coefficient is used. The purpose of applying different coefficients is to modulate the tradeoffs in symptoms.

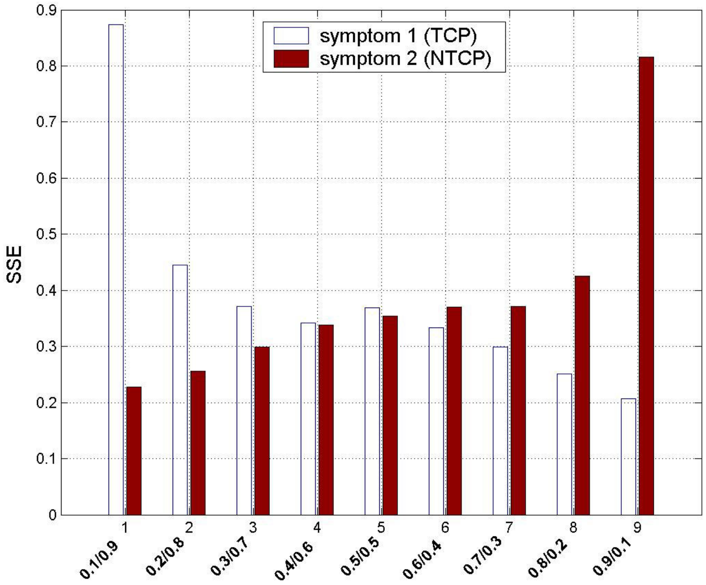


Figure 38: Compare SSE of different coefficient pairs

The relationship between predicted outputs (follow-up symptoms 1 and 2) from the PM and their corresponding targets using the coefficient (0.1, 0.9) is illustrated in Figures 40. The left side of Figure 39 shows the relationship of the output (symptom 1) and its desired target when a coefficient in the first objective function is 0.1. The right side of Figure 39 displays the output (symptom 2) and its corresponding target relationship when the coefficient is 0.9 in the second objective function. Likewise, Figures 40 illustrates output and target relationships when the coefficient (0.9, 0.1) is used in the objective function. The symptom, which is employed with a higher coefficient, signifies a higher correlation between the output and its target than the lower coefficient symptom.

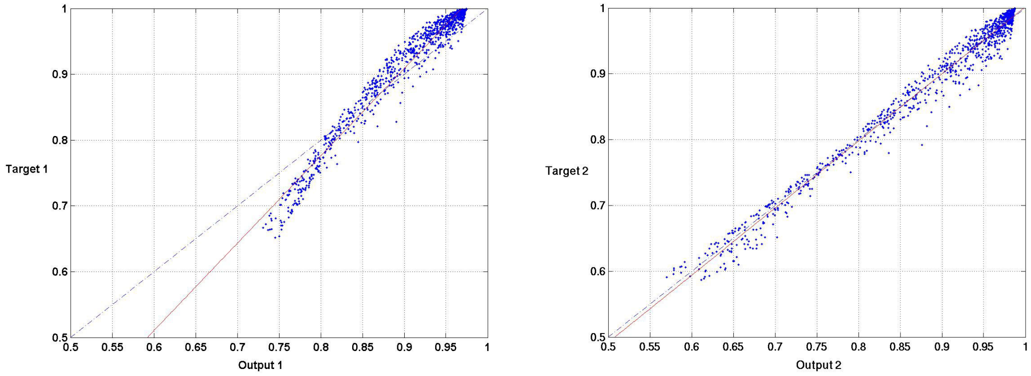


Figure 39: Coefficient (0.1,0.9)

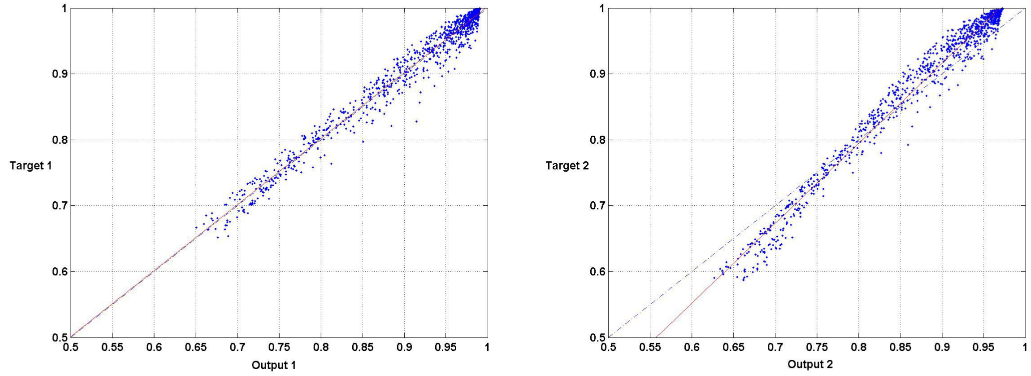


Figure 40: Coefficient (0.9,0.1)

To assess whether the slope of each graph is equal to 1 (H_0: $\beta_1 = 1$), a regression test with a significant level of 0.05 ($\alpha = 0.05$) was used to test the hypotheses of these 4 (graphed) cases. The details about the regression test and formula can be found in many statistics texts such as [Kleinbaum, Kupper, and Muller, 1998]. The hypotheses test results can be summarized as follows:

- Significant when the first output unit of (0.1,0.9) is employed with a 0.1 weight.
- Not significant when the second output unit of (0.1,0.9) is employed with a 0.9 weight.
- Not significant when the first output unit of (0.9,0.1) is employed with a 0.9 weight.
- Significant when the second output unit of (0.9,0.1) is employed with a 0.1 weight.

Here, not significant means that the slope is not different from 1 (equal to 1), while significant means the slope is different from 1. The results are consistent with the graphical illustrations.

The results from Figures 38-40 indicate that higher correlations are achieved when higher coefficients are applied to the objective function. These results can be applied in situations where a clinician wants to emphasize the prediction of one symptom over the other such that a major symptom will be given a higher coefficient than a minor one.

The performance of the PM can also be measured by comparing the error function computed by the PM, trained with a coefficient (0.5, 0.5), with the error computed from the target function. The focus here is on the coefficient (0.5, 0.5) because it will be employed later in Section 4.3.2 Study 2b. Figure 41 shows the error function of the network and target function. The solid curve represents the error function from the target function, while the dashed-line curve denotes the error function from the network. The results from the graph indicate that the error function of the PM is very close to the target function. This means that the network approximates the target function quite well.

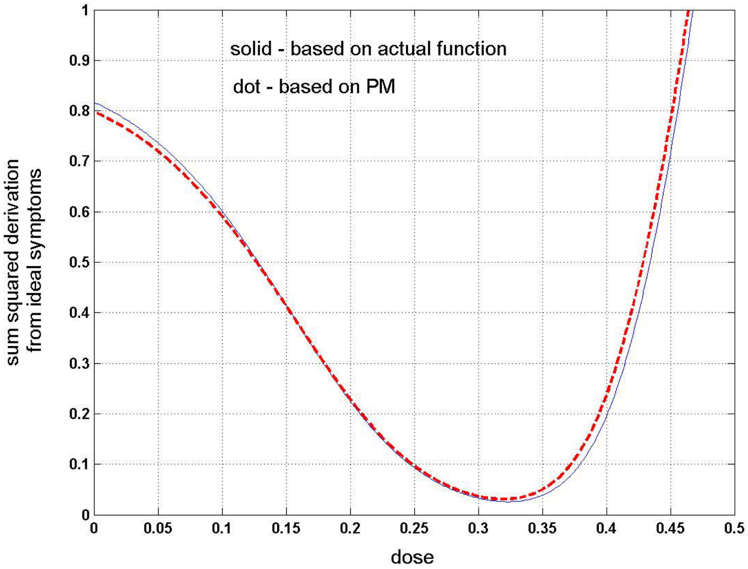


Figure 41: Compare PM and target error functions

4.3.2 Study2b: Analysis of coefficients (B_1, B_2) in the treatment network when both objective function terms are quadratics

4.3.2.1 Methods

This study explores different coefficients in the TN when both objective functions are quadratic. Training is divided into two stages: training the patient network and training the treatment network. The coefficient for the patient network (A_1, A_2) was fixed at (0.5, 0.5). Once the patient network was trained, the weights from that network are frozen and employed for training the treatment network. Different coefficient pairs (B_1, B_2) on the objective function are used in training the treatment network. Initial weights, learning rate, momentum, number of hidden units, and activation functions for both networks are determined empirically. This analysis can be used in the situation where a clinician wants to optimize one treatment over the other one. This allows the clinician to adjust the treatment (dose) according to the health status in each patient.

4.3.2.2 Results

Since the ideal doses could not be calculated directly in this experiment, the follow-up symptoms, which can be computed from the target function (using equation [11] and the conditions in Figure 35), will be used for analyzing the results. As a reminder, the TN is trained to find treatment values (doses) that can optimize the symptoms. Therefore, if the recommended doses from the TN are close to the optimum doses, then the symptoms should also be minimized. In other words, the follow-up symptoms generated using predicted doses from the TN should be along the ideal target curve. (To compute follow-up symptoms of the TN, the predicted TN doses, together with the corresponding initial symptoms, will be given to the target function).

The effects of different coefficients on the cost function of the TN can be demonstrated in Figure 42. The graph displays the relationship between TCP and NTCP. The curves on the graph represent ideal TCP and NTCP relationships for a male patient 55 and 65 years of age. Symptoms from the predicted doses of the TN are denoted with the symbol ‘*’. The lowest ‘*’ point of each curve is derived from the predicted dose of the TN when the coefficient is (0.1, 0.9). The next upward point of each curve is a consequence of applying a coefficient of (0.2, 0.8) and so on until the top most point ends with a coefficient of (0.9, 0.1).

It can be seen that symptoms from the recommended doses of the TN lie along the ideal curves. In addition, different coefficient pairs have different effects. The results from the graph suggest that the higher the coefficient, the lower the symptom. This analysis can be useful when a clinician wants to optimize one treatment over the other treatment.

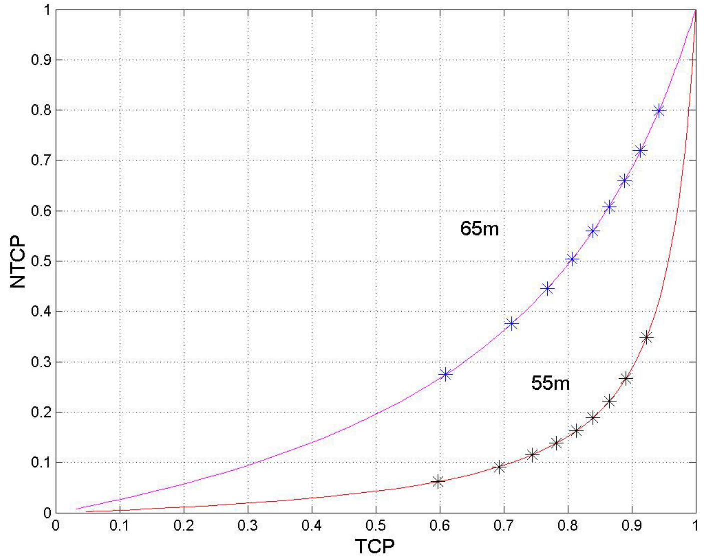
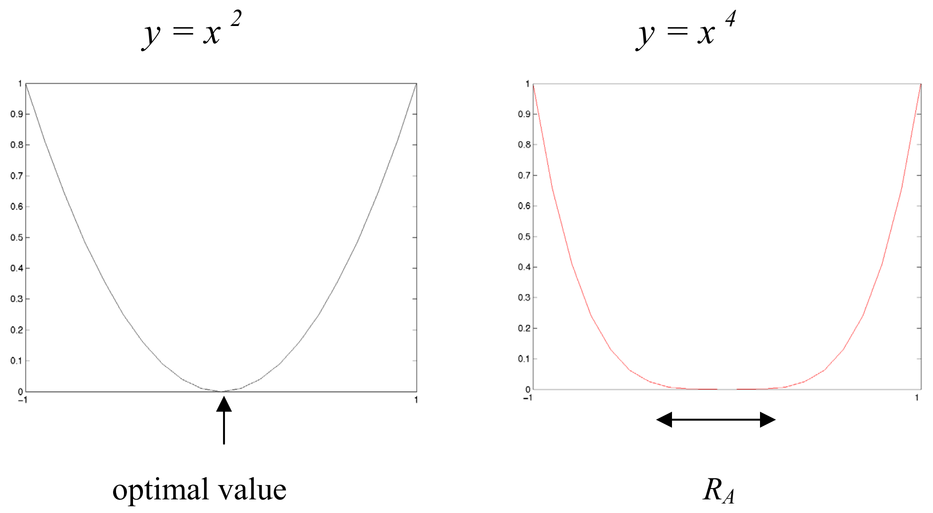


Figure 42: Effects of cost function coefficients on TN

4.3.3 Study 2c: Analysis of the treatment network when one objective function term is quadratic and the other objective function is a higher exponent (i.e. degree 4).

In general, the quadratic cost function is the standard used in back-propagation training. However, a higher exponent may be more appropriate in situations where the target value has a long acceptable range over a finite interval. See the following figures for a comparison between the quadratic and quartic (exponent 4). It can be seen that the higher exponent yields an acceptable range $R_{A.}$ The higher exponent curve increases much more suddenly than the quadratic, acting much like a threshold function at the bound of R_A . A parameter d^* is introduced to scale the argument to the function, such that the threshold-like behavior of the cost function occurs when the dose is in the neighborhood of d^*.



4.3.3.1 Methods

Training is divided into two stages: training the patient network and training the treatment network. The coefficients for the patient network, (A_1, A_2), are fixed at (0.5, 0.5). Once the patient network is trained, the weights from that network are frozen and employed for training the treatment network. Different coefficient pairs (B_1, B_2) on the objective function, one with quadratic and the other with a higher exponent, are used in training the treatment network. Initial weights, learning rate, momentum, number of hidden units, and activation functions for both networks, are determined empirically. The objective function for the treatment network to be optimized is shown in Equation [14].

$$B_1 \sum_{\alpha}^{N_{tr}} (s_1 * - r_{PM_1} (a^\alpha, g^\alpha, tz^\alpha, r_{TN}(d^\alpha)))^2 + B_2 \sum_{\alpha}^{N_{tr}} (s_2 * - r_{PM_2} (a^\alpha, g^\alpha, tz^\alpha, r_{TN}(\frac{d^\alpha}{d*})))^4 \quad [14]$$

4.3.3.2 Results

Different exponents (TCP with exponent 2 and NTCP with exponent 4) on the cost function of the TN network are investigated. In addition, threshold values of NTCP ($d*$ in equation [14]) are set at 0.5, 0.4, 0.3, and 0.2, with coefficient (B_1, B_2) equal to (0.9, 0.1). Figure 43 illustrates TCP and NTCP relationship for a male 55 and 65 years of age. The symbol '*' on the graph denotes predicted symptoms from the TN when the threshold of NTCP is set at 0.5, 0.4, 0.3, and 0.2. The upper most points '*' on both curves are the outcomes of employing a 0.5 threshold, followed by the results of using 0.4, 0.3, and 0.2 threshold values, respectively.

It is clear from the graph that when the threshold is 0.5, NTCP symptom values do not exceed 0.5 for both Male 55 and Male 65. Likewise, when threshold values are 0.4, 0.3, and 0.2, NTCP symptoms do not exceed the specified thresholds. Therefore, the specified threshold on the higher exponent (NTCP) acts as an upper limit of the NTCP symptom. This analysis allows a clinician to specify the upper limit of NTCP that patients can tolerate. For example, how many lost normal cells that can be endured by a patient.

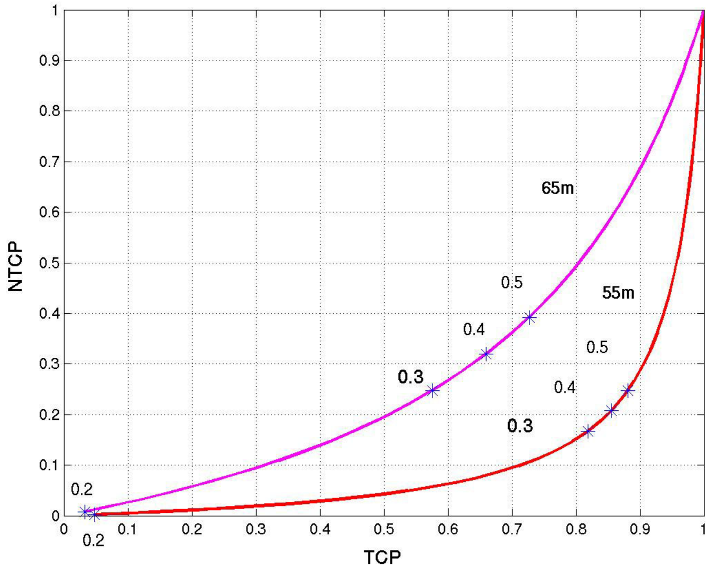


Figure 43: Effects of different powers and thresholds on the TN

To further explore the effects of a threshold on higher exponents compared with quadratics, the TN is trained with a fixed threshold of 0.4. As in the PM of study 2a, coefficient pairs (0.1,0.9), (0.2,0.8), …, (0.9, 0.1) are utilized, where the first component of a coefficient pair is used in the cost function of symptom 1 (TCP), and the second component in a coefficient pair is used in the cost function of symptom 2 (NTCP).

Figure 44 demonstrates the effect of a higher exponent (e.g. 4) in the NTCP cost function that is trained with different TCP coefficients for a male 55 years of age. The x-axis represents TCP coefficients starting from 0.1-0.9. The y-axis displays the NTCP and doses. The symbol '*' in the graph denotes the NTCP value, while the symbol ‘o’ signifies dose values that correspond to the TCP coefficients.

Figure 45 illustrates the effect of a quadratic in the NTCP cost function. The graph in Figure 45 gradually increases when the coefficient increases, while the graph in Figure 44 increases sharply at a certain coefficient. The comparison of results from Figures 44 and 45 indicate that the threshold in the higher exponent acts as a hard threshold.

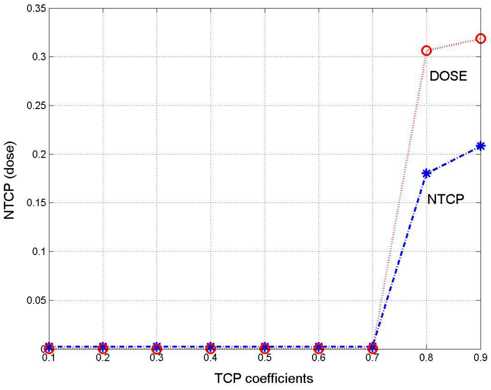


Figure 44: M55, exponent 4

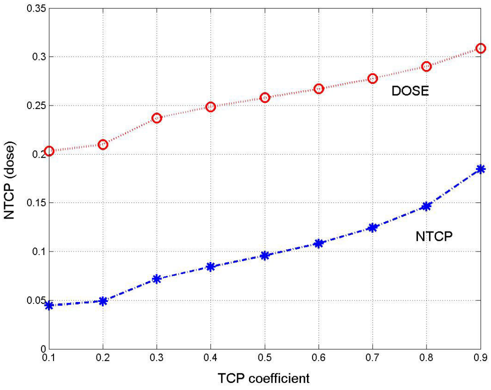


Figure 45: M55, with quadratic

Figure 46 shows results of a higher exponent on males 65 years of age. Figure 47 displays the results of a quadratic on males 65 years of age. Again, it can be seen that the NTCP and dose values in the quadratic are gradually increased but NTCP and dose values in the higher exponent are increased sharply at a certain coefficient.

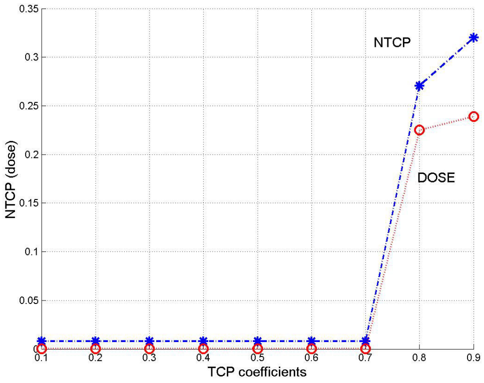


Figure 46: M65, exponent 4

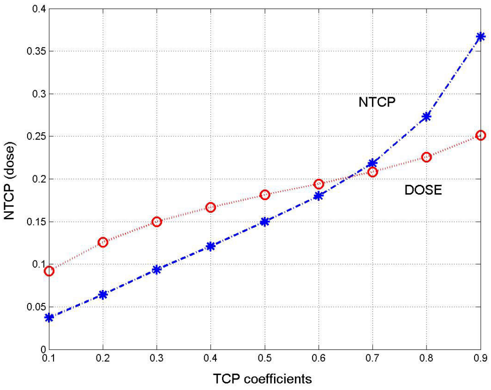


Figure 47: M65 quadratic

To further ascertain that conclusion, a comparison of exponent 4 and quadratic for males 45 years of age are illustrated in Figures 48 and 49, respectively. The results from these figures are consistent with Figures 44 through 47.

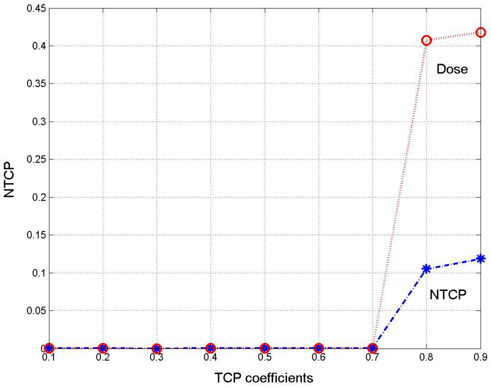


Figure 48: M45, exponent 4

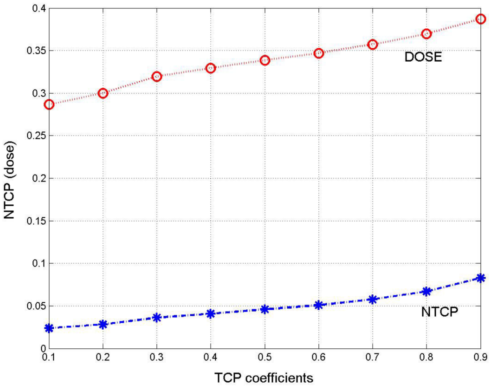


Figure 49: M45, quadratic

4.4 Study 3: multi-objective functions using two doses where there are dose interactions

This study focuses on multi-objective functions, multi-treatments, and the interaction between those treatments.

A drug-drug interaction occurs when one drug alters the way in which another drug affects the body. The consequence of such interactions can be harmful, such as increasing the toxicity of the drug and/or reducing its efficacy. They can also be beneficial. Although drug interactions with the potential to threaten lives are rare, the decrease in drug effectiveness could result in complications and unnecessary costs.

It would be easier to determine the response of a specific drug and any side effects if a person receives one drug at a time. In reality, and for many reasons, one person might receive multiple medications at once. Combinations of drugs are important for treating such conditions as AIDs and cancer. Also, many people who have more than one disorder may need different drugs to treat each condition.

According to [MeRec Bullentin, 1999], mechanisms of drug interaction involve pharmacokinetic and pharmacodynamic interactions. Pharmacokinetic interactions affect the processes by which drugs are absorbed, distributed, metabolized, or excreted in the body [Stockley, 1996]. In short, absorption is the process in which a chemical agent enters the blood. Distribution is the process in which a chemical agent is distributed throughout the body. Metabolism is the process in which the chemical agent is broken down or modified by the organism, usually via enzymes. Finally, excretion is the process in which toxicants are eliminated from body. Pharmacodynamic interactions result in an increase (synergism) or decrease (antagonism) response of one or more drug in the additive or opposing direction. Here, the focus is one the latter type of interaction. Understanding the potential of drug interactions can increase the chance of a successful therapeutic response.

The subjects of drug interactions can be varied and complex. However, to minimize the controversy, the concern here is on *simple* drug interactions. Simple interactions are referred to as interactions between an active and inactive drug. The interaction can result in an increase or decrease of the effect of another drug. A drug action that increases the effect produced by the other drug is called *synergistic action.* On the other hand, a drug action that counteracts or decreases the action of another drug is called *antagonistic action* [Poch, 1993].

4.4.1 Methods

Generating data for multi-objective functions and drug interaction:

Each record (α) consists of seven variables: age (a), gender (g), treatment 1 (d_1), treatment 2 (d_2), and the results (r_1, r_2, r_3). The first two variables represent the initial state of the patient, the next two variables denote treatments, and the last three variables are meant to represent increases in symptoms. In this study, there is no representation of the initial symptoms.

The result symptoms (r_1, r_2, r_3) are computed using the following equations:

$$r_1 = e^{-(\alpha_{11} d_1 d_2 + \alpha_{21} d_1)} \qquad [15]$$

$$r_2 = e^{-(\alpha_{12} d_1 + \alpha_{22} d_2 + \alpha_{32} d_1 d_2)} \qquad [16]$$

$$r_3 = e^{-(\alpha_{13} d_1^{\;2} d_2 + \alpha_{23} d_1 - \alpha_{23} d_2)} \qquad [17]$$

In [15], α_{11} is a constant, while α_{21} is a function of age (5/age).

In [16], α_{12} and α_{32} are constant, while α_{22} is a function of gender (male 0.6, female 0.8).

In [17], α_{13} is a function of age (300/age), while α_{23} is a function of gender (male 0.2, female 0.3)

These functions were derived from the shape of curves graphed in an experiment by [Krishan et.al.2000]. The following graphs (Figures 50, 51, and 52) show the dose and toxicity effect relationships of one drug (i.e. treatment 1) and a combination of drugs (i.e. treatment 1 and treatment 2) in three different cancer cell types. The toxicity effects and dose relationships of follow-up symptom 1, follow-up symptom 2, and follow-up symptom 3 are illustrated in Figures 50, 51, and 52, respectively.

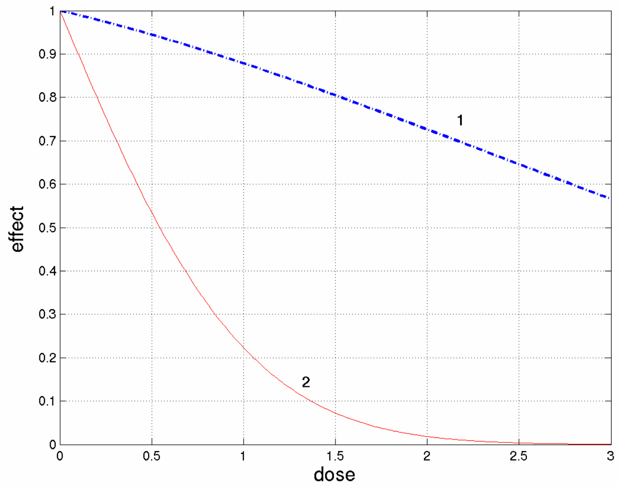


Figure 50: Simulated dose and toxicity effect relationship of one drug and a combination of drugs in a selected cancer cell type (for symptom 1)

In Figure 50, curve 1 (dashed-line) shows the dose-effect of treatment 1 alone. Curve 2 (solid line) displays the dose-effect of treatment 2 with the presence of treatment 1. It can be seen that the combination of treatment 1 and treatment 2 interact in such as way that the toxicity effect is lower as compared with treatment 1 only. In other words, with the same level of dose, the combination of treatments gives a lower toxicity level than when using treatment 1 alone.

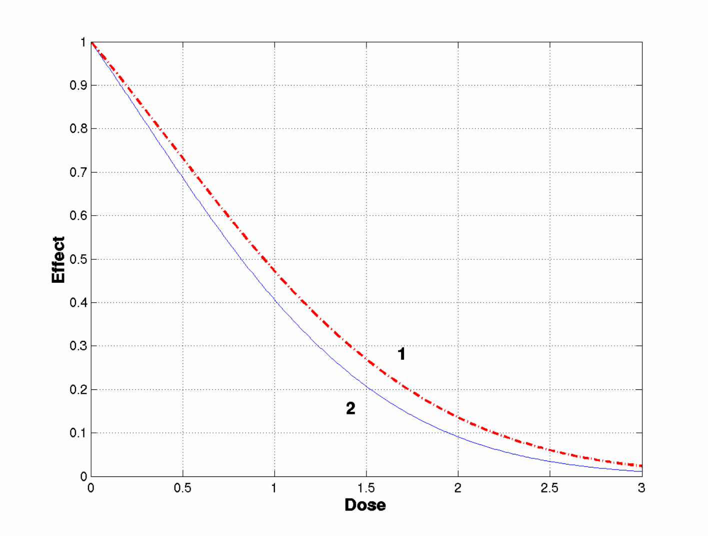


Figure 51: Simulated dose-effect relation of one drug and a combination of drugs in a selected cancer cell type (for symptom 2)

In Figure 51, curve 1 (dash-line) shows the dose-effect of treatment 1 alone. Curve 2 (solid line) displays the dose-effect of treatment 2 with the presence of treatment 1. It can be seen that there is little difference in the toxicity level when using treatment 1 alone, as compared to the combination of treatment 1 and treatment 2.

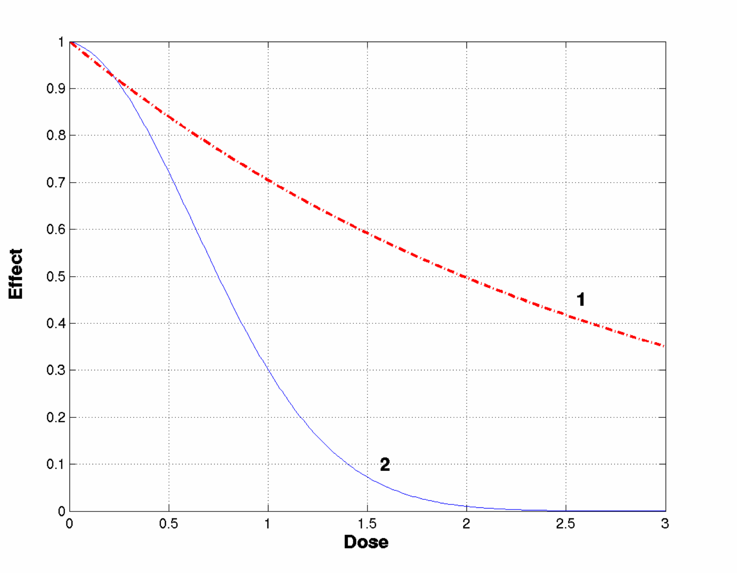


Figure 52: Simulated dose-effect relation of one drug and a combination of drugs in a selected cancer cell type (for symptom 3)

In Figure 52, curve 1 (dash-line) shows the dose-effect of treatment 1 alone. Curve 2 (solid line) displays the dose-effect of treatment 2 with the presence of treatment 1. It should be noted that curve 2 is very bent at the beginning. Therefore, the quadratic form is used in equation [17]. It can be seen that the combination of treatment 1 and treatment 2 interacts in such as way that the toxicity effect is lower when compared with treatment 1 only. In other words, with the same level of dose, the combination of treatments results in a lower toxicity than when using treatment 1 alone.

As a result, the patient network must learn how to predict the follow-up symptoms and the treatment network must learn how to adjust treatment 1 and treatment 2 to get the lowest toxicity level for all follow-up symptoms.

The objective function in the patient network is:

$$\sum_{i}^{3} \sum_{\alpha}^{N_{tr}} A_{i}(s_{f_{i}} - r_{PM_{i}}(a^{\alpha}, g^{\alpha}, d_{1}^{\alpha}, d_{2}^{\alpha}))^{2} \qquad [18]$$

Where symbol A represents a coefficient for the relative importance factor of the objective function. Symbol s_{f_i}, when i is {1, 2, 3}, denotes the following-up symptoms ($s_{f_1}, s_{f_2}, s_{f_3}$), respectively, and r_{PM_i} is the output of the patient network

The objective function in the treatment network is:

$$\sum_{i}^{3} \sum_{\alpha}^{N_{tr}} B_i (s_i * - r_{PM_i}(a^{\alpha}, g^{\alpha}, r_{TN_i}(d_1{}^{\alpha}, d_2{}^{\alpha})))^2 \qquad [19]$$

Where symbol B represents a coefficient for the relative importance factor of the objective function. Symbol s_i^* when i is {1, 2, 3} represents the desired symptom value, and r_{TN_i} is the output of the treatment network.

4.4.2 Results

4.4.2.1 Performance Analysis of the patient network with different coefficients.

The PM consists of three output units, each unit corresponding to a symptom: r_1, r_2, and r_3. Each symptom can be treated with different weights. For example, a coefficient of (1,0,0) means that the first symptom is assigned with the first coefficient weight of 1, while the second and third symptoms are assigned with a weight of 0. Since there are many possible combinations of these three result symptoms, the coefficients that are selected to be used in the experiments consisted of: (0.33,0.33,0.33), (0.5,0,0.5), (1,0,0), (0.5,0.5,0), (0,1,0), (0,0.5,0.5), and (0,0,1). These coefficients are applied to the objective (cost) function of the PM network.

Figures 53-55, respectively, show the contour plot of result symptoms 1, 2, and 3 of the network compared with their corresponding target symptoms for a 45 year old male when the coefficient is (0.33,0.33,0.33).

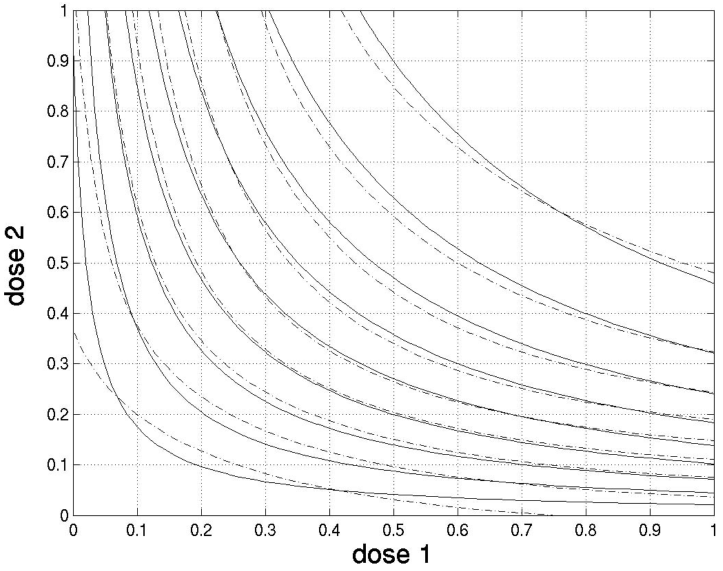


Figure 53: Contour plot for result symptom 1 of target function compared with the PM

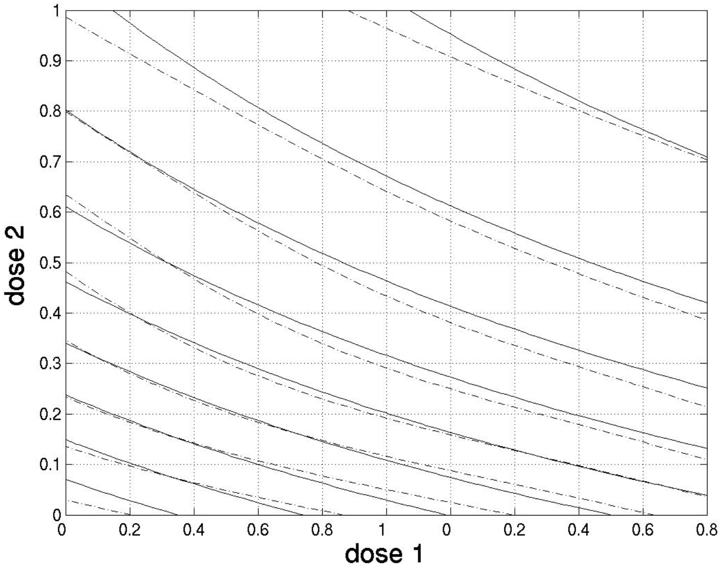


Figure 54: Contour plot for result symptom 2 of target function compared with the PM

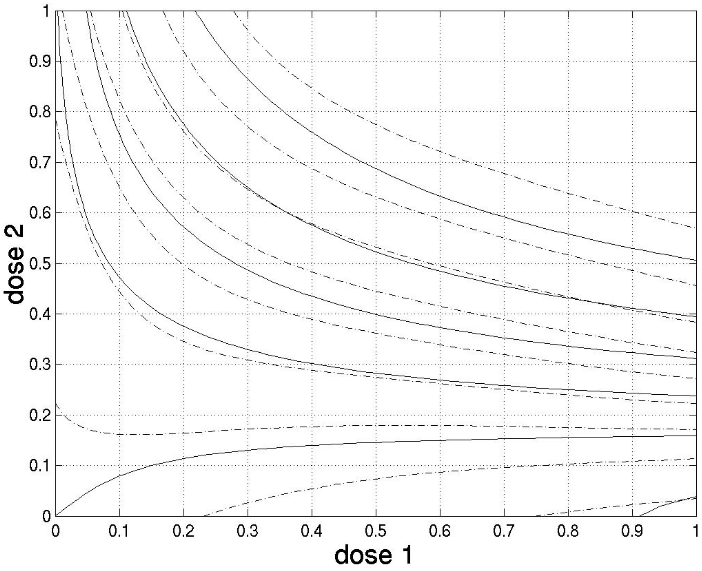


Figure 55: Contour plot for result symptom 3 of target function compared with the PM

The solid lines in Figures 53-55 represent contour plots of the target values, while the dashed lines show the contour of the network. The contour plots of the PM are very close to that of the target function.

Of those 7 different coefficients mentioned earlier, the error of the PM cost functions is calculated. Doses that result in the minimum of the cost function corresponding to each coefficient are determined. These doses should be the optimal doses because the cost function values are the minimum. It is important to note that the first and third symptoms are assumed to be bad cell types while the second symptom is presumed to be a good cell type. The comparison of the optimal doses of the PM and ideal functions of a 45 year old male is shown in Figure 56.

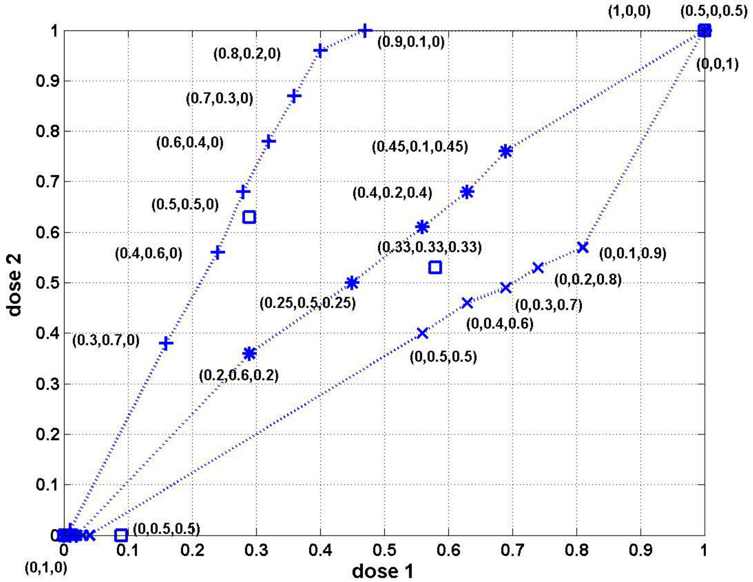


Figure 56: Compare optimal doses from the PM and ideal functions

The coefficients of the ideal functions displayed in Figure 56 are divided into three cases. In the first case, the third coefficient is fixed at zero: (0.1,0.9,0), (0.2,0.8,0),…, (0.9,0.1,0), (1,0,0). In the second case, the first and third coefficients are equal, such as: (0,1,0), (0.05,0.9,0.05), (0.15,0.7,0.15), (0.2,0.6,0.2), (0.25,0.5,0.25), (0.33,0.33,0.33), (0.4,0.2,0.4), (0.45,0.1,0.45), and (0.5,0,0.5). For the third case, the first coefficient is fixed to zero: (0,1,0), (0,0.9,0.1),…, (0,0.1,0.9), and (0,0,1).

The optimal doses of the PM are illustrated with a square symbol. The optimal doses of the ideal functions are displayed with the symbol '+' when the third coefficient is fixed to zero; with the symbol '*' when the first and third coefficient are equal; and finally, with the symbol 'x' when the first coefficient is fixed to zero.

It can be seen that the optimal dose calculated from PM and the ideal functions are the same when the coefficients employed are (1,0,0), (0.5,0,0.5), (0,0,1), and (0,1,0). The optimal doses of the PM and ideal functions are very close in the case of the coefficients (0.33,0.33, 0.33), and (0.5,0.5,0). However, in the case where coefficient is (0,0.5,0.5) the optimal dose of the PM is quite far off the optimal dose of the ideal function. The comparison of the surface plot cost function for both the ideal and the PM for three latter coefficients will be analyzed further later in this section.

Second-ordered polynomials (quadratic response surfaces) are generally used as a local approximation to the true input/output relationship in an empirical modeling approach such as Response Surface Methodology (RSM). (More detail about RSM can be found in Section 2.7).

According to [Myers and Montgomery, 2000], the second-order model is widely used in RSM because of its flexibility. There are several different surfaces that can be generated by second-order models (see example in Figure 15). It is a good approximation to the true response surface. Additionally, substantial practical experience suggests that "second-order models work well in solving real response surface problems" [Myers and Montgomery, 2000]. Quadratic surfaces have just one minimum; hence, they do not have the potential ambiguity of more complex function (e.g. the PM).

Let the quadratic response surface (QRS) model be defined as a 2nd order polynomial fit to the same data fit by the PM. It will be used as a benchmark to evaluate the PM performance. The QRS model used here is in the form:

$$Y = \beta_0 + \sum_{j=1}^{K} \beta_j X_j + \sum_{j=1}^{K} \beta_{jj} X_j^{\ 2} + \sum_{j=2}^{K} \sum_{i=1}^{j-1} \beta_{ij} X_i X_j$$

The QRS is generated using the Matlab Statistics Toolbox to approximate the unknown parameter β_0, β_1,.., β_{14}. of the above model. Therefore, there are three fitting nonlinear models corresponding to the three result symptoms in this study. The models are then used to calculate the cost function of the seven different coefficients in the PM. Then, the corresponding optimal doses for each coefficient are determined.

Figure 57 displays optimal doses of the ideal function, the PM, and the QRS for a male 45 years old.

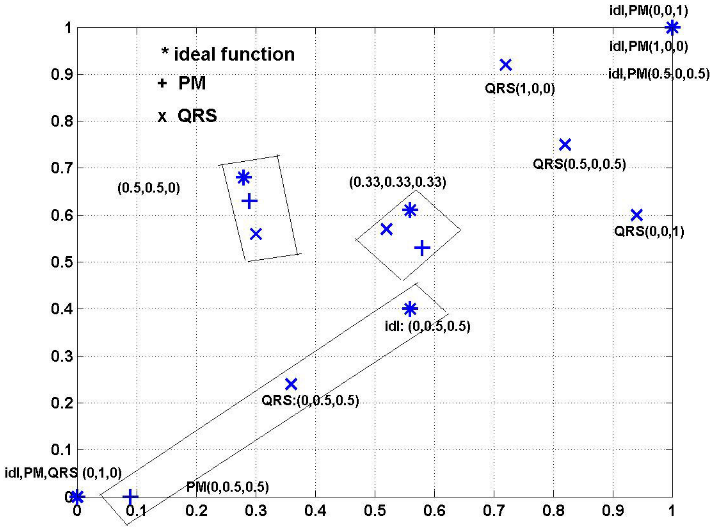


Figure 57: Compare optimal doses from the ideal function, PM and the QRS

From Figure 57, '*idl*' represents the ideal function with symbol '*', 'PM' is the PM network using the symbol '+'. Finally, 'QRS' is the QRS with the symbol 'x'. It can be seen that the optimal doses from the PM are the same as the ideal functions when the coefficients used are (0,1,0), (0,0,1), (1,0,0), and (0.5,0,0.5), while the optimal doses from QRS are the same as ideal function only when the coefficient is (0,1,0).

It is obvious that optimal doses from PM result in lower cost function errors than the QRS when the coefficients employed are (0,0,1), (1,0,0), and (0.5,0,0.5). The PM and QRS are comparable in the case where the coefficient is (0,1,0). Thus, the comparison in the cost function and surface plot of the ideal function, PM, and QRS will be focused merely on the coefficients (0.33,0.33,0.33), (0.5,0.5,0), and (0,0.5,0.5).

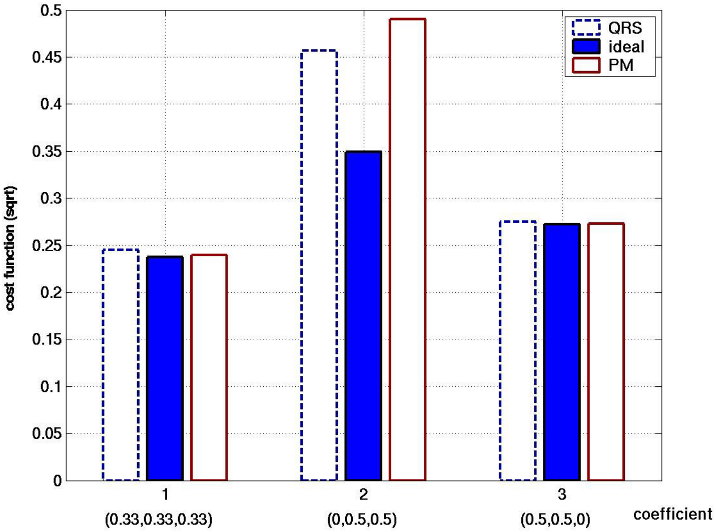


Figure 58: Compare cost function error of the ideal function, PM and the QRS

The cost function comparisons of the QRS, ideal function, and PM are shown in Figure 58. The dashed bar on the left is the QRS, the filled bar in the middle is the ideal function, and the unfilled bar on the right is the PM. The cost functions of the optimal doses from ideal function are expected to be the lowest for all coefficient cases. The cost functions of the PM are lower than the QRS when the coefficients are (0.33,0.33,0.33) and (0.5,0.5,0). However, the QRS cost function is lower than the PM merely in the case where the coefficient is (0,0.5,0.5).

It is interesting to further explore the surface plots of each model for comparison. Figures 59-67 compare the surface plot of the cost function calculated from ideal functions, PMs, and QRSs.

Figures 59-61 display the surface plots of the cost function of the ideal function, PM, and QRS for a 45 year old male when the coefficient is (0.33,0.33,0.33).

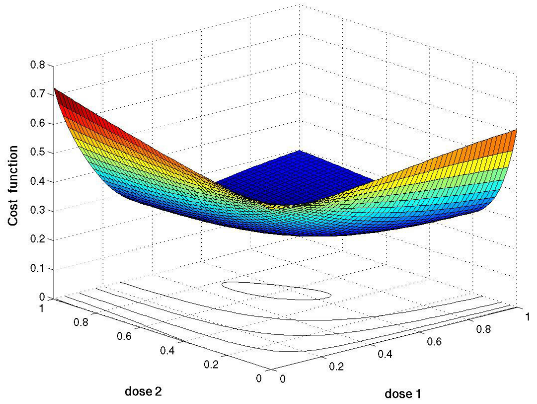


Figure 59: surface plot of the ideal function with coefficient (0.33,0.33,0.33)

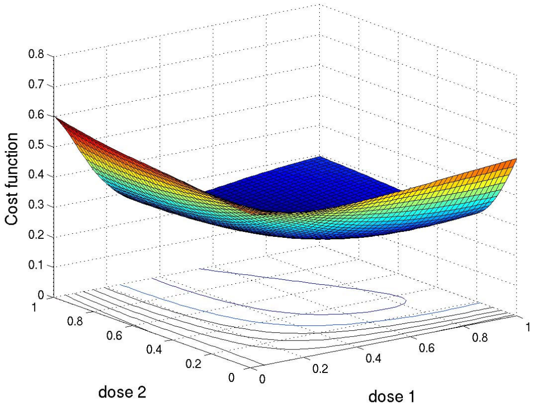


Figure 60: surface plot of the PM with coefficient (0.33,0.33,0.33)

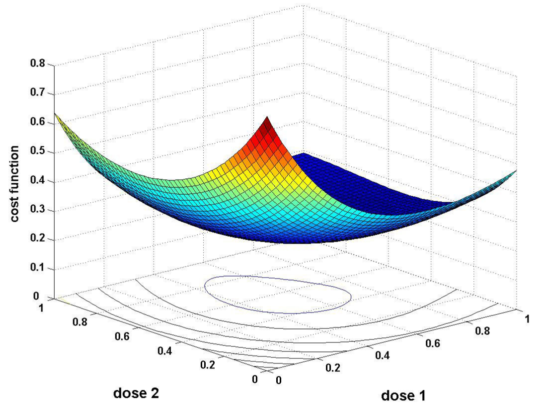


Figure 61: surface plot of the QRS with coefficient (0.33,0.33,0.33)

Figures 62-64 display the surface plots of the cost function of the ideal function, PM, and QRS for a 45 year old male when the coefficient is (0.5,0.5,0).

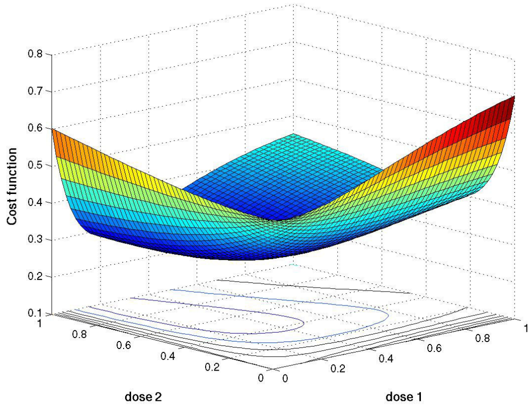


Figure 62: surface plot of the ideal function with coefficient (0.5,0.5,0)

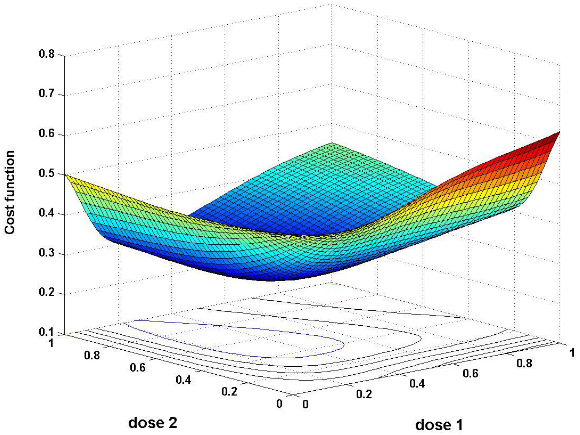


Figure 63: surface plot of the PM with coefficient (0.5,0.5,0)

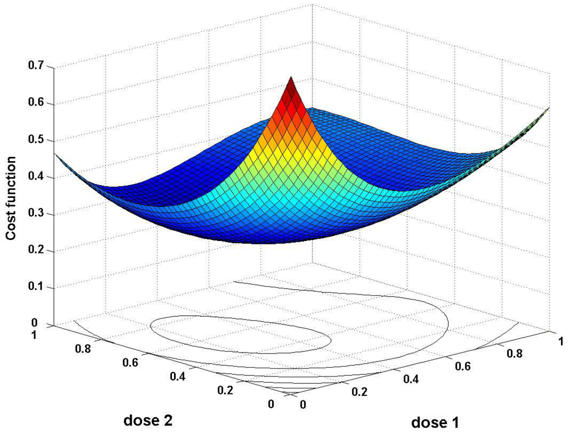


Figure 64: surface plot of the QRS with coefficient (0.5,0.5,0)

Figures 65-67 display the surface plots of the cost function of ideal function, PM, and QRS for a 45 year old male when the coefficient is (0,0.5,0.5).

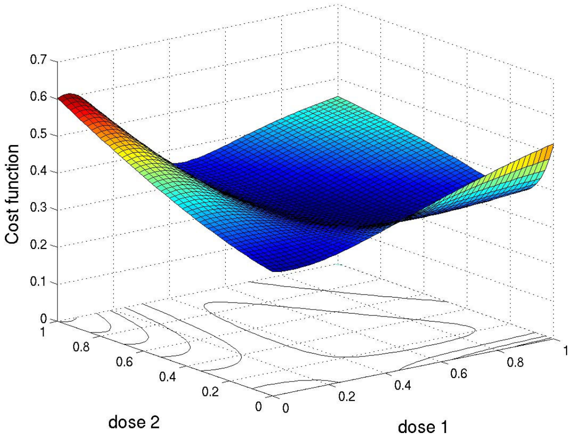


Figure 65: surface plot of the ideal function with coefficient (0,0.5,0.5)

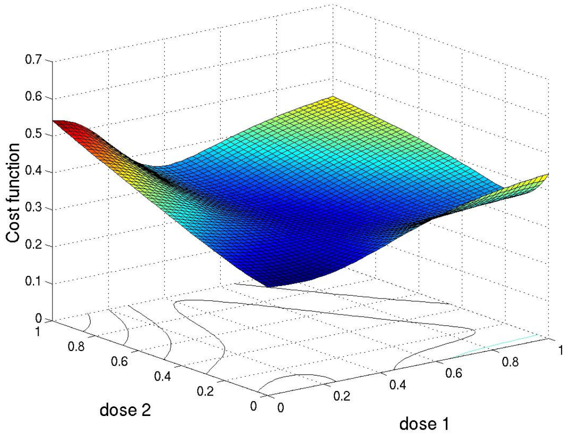


Figure 66: surface plot of the PM with coefficient (0,0.5,0.5)

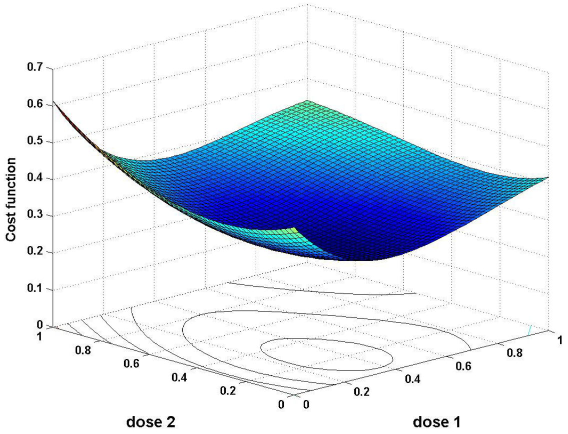


Figure 67: surface plot of the QRS with coefficient (0,0.5,0.5)

The shapes of the surface plots of the PM are very similar to the ideal function in these three cases. However, the surface of coefficient (0,0.5,0.5) is particularly interesting. From Figure 66, it appears to have two flat regions; one in the middle of the surface, where the optimal dose of ideal function occurs (0.56,0.4), as well as another near the origin (0.09,0), where the optimal dose of the PM is found. Although 30 additional simulations were run with different initial weights and learning rates, the optimal doses calculated from the PM were always located near the origin. This behavior requires further discussion.

Back-propagation performs a gradient decent search in a weight space towards a global minimum. Unfortunately, the solution space in most problems may consist of numerous local minima that may not be the best overall solutions. Local minima are common problems for descent-type search methods, which generally can be approached by training the network with different initial weights, and comparing the convergent states.

In this framework, the network was employed to solve non-linear problems. It is possible that there may be several feasible solutions for those types of problems. For example, any combination of drug A and drug B within the range of 10-15 mg may be nearly equivalent, and very effective, with 11.2 mg of A and 14.5 mg of B being optimal. This would correspond to a "*flat region*" in the cost function. Another situation is illustrated by the following example. The value of the cost function plotted in dose space is bimodal. Suppose that local minima are found at two points in dose space: A=10mg, B=3mg and at A=5mg, B=9mg. Finding the global minimum (10,3) depends on the initial state. Thus, running several networks with different initial states is an approach worth considering when the number and nature of local minima is unknown.

Consider the surface plot of PM in Figure 66, it appears to have two flat regions: one in the middle of the surface, where the optimal dose of ideal function occurs (0.56,0.4), as well as the others near the origin (0.09,0), where the optimal dose of the PM is found. Although 30 additional simulations were run with different initial weights and learning rates, the optimal doses calculated from the PM were always located near the origin. The cost functions between these two locations of PM were compared. It shows that the location near the origin has a lower cost function. As a result, optimal doses from PM are always found near the origin.

One of the possible explanations why optimal results from the PM are always found near the origin, instead of near the optimal of the ideal functions, is that the local minima in dose space are not necessarily local minima in weight space. For instance, if the weights of two different output nodes are exchanged, it is still possible to get the same solution in dose space. Thus, no matter how many times the network is trained it always comes down to the same values. In the event that there are several local minima in "*weight space"*, multiple simulations with different initial weights are advisable.

4.4.2.2 Performance Analysis of the treatment network

The weights of the patient network trained with coefficient (0.33,0.33,0.33) in the previous experiment are employed for patient network training. After the TN is trained, the suggested doses from the six age-gender combinations (male and female; 25, 45, and 65 years of age) are determined. Then, the cost functions from the TN dose suggestions are calculated and compared with the cost functions of the ideal and QRS. The results are shown in Figure 68.

From Figure 68, the categories of age and gender are depicted on the X-axis consisting of a 25 year old male (25M), 45 year old male (45M), 65 year old male (65M), 25 year old female (25F), 45 year old female (45F), and 65 year old female (65F). The Y-axis displays the cost function. In each category, the left most bar is the QRS shown with a dashed bar, the middle bar is the ideal function shown with a filled solid bar, and the unfilled bar on the right is the TN. For all categories, the ideal functions appear to have the lowest cost function values. Cost functions of the TN in all categories are lower than the QRS.

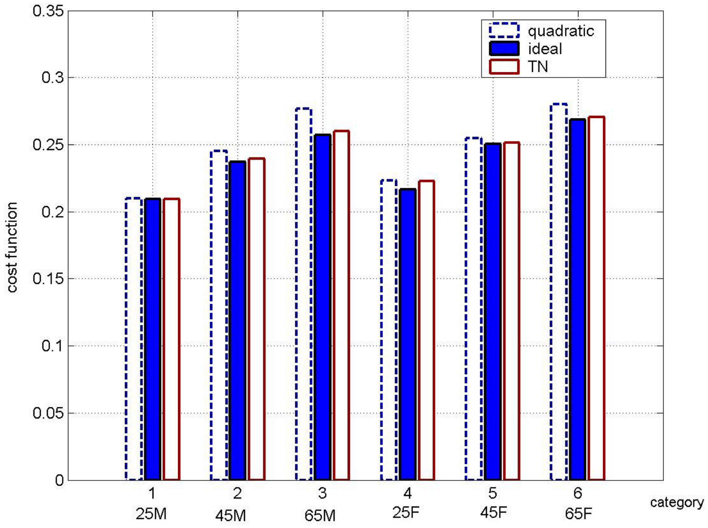


Figure 68: Compare cost function of the ideal function, TN and QRS

5. Summary, Conclusion and Future Research Directions

Summary

The distal learning technique was introduced here to the medical domain to find optimal doses using large sets of patient data. The approach is a two-network system: one network, called a patient model (PM), is used to predict the outcome of the treatment, while the other, called a treatment network (TN), is used to optimize the predicted outcome.

Three set of studies were performed using artificial data (patient states, treatments and subsequent states) generated using mathematical models:

- Initial studies explored just one objective function (one symptom), two doses with and without interaction between these doses. In addition, the effects of noise and different sample sizes, when there is an interaction between doses, were explored.

- Subsequently, multi-objective functions (multiple symptoms) using one dose were examined. Solutions here are based on finding the best possible dose combinations that optimize different objectives simultaneously. The problem becomes more complicated when there are conflicts between those objective functions. Chemotherapy and radiotherapy, for example, are medical applications that deal with multi-objective criteria. The focus is to design drug regimen strategies to ensure that tumor cells are killed at the desired rate, but at the same time, should not cause unacceptable toxicity to the healthy cells. The data used in this experiment were inspired by a radiotherapy problem. Different coefficients in PM, different coefficients in TN cost function and different exponents (quadratic and quartic) in TN were investigated.

- Finally, multi-objective functions (multiple symptoms), using multiple doses where there is an interaction between doses were studied. The concern here is on simple drug interactions. Drug combinations are important in treating conditions such as cancer and AIDS. The functions used were derived from experimental graphs of [Krishan et.al. 2000] as described in Section 4.4. In addition, the performance of the networks (PM and TN) and quadratic models were evaluated.

Conclusion

A number of researchers, such as Stengel et.al. [2002] Illiadis et.al. [2000], utilized underlying knowledge in pharmacokinetics (PK) and pharmacodynamics (PD) for treatment optimization. In short, PK is “what the body does to the drug” [WebMD] and PD is “what the drug does to the body” [WebMD]. More information about PK/PD can be found in Section 2.6. The approach employed in this study, however, ignores all biochemical mechanisms and is purely based on patient statistics.

The effect of sample size was examined. Smaller sample sizes result in higher error. The average error tends to be lower in the larger sample size. For every sample size, the average error with noise present is slightly higher than when there is no noise.

Different coefficients were employed in the PM to modulate the tradeoffs in symptoms. The relative importance of symptoms can be adjusted by applying weights in the objective functions. The network achieves high accuracy on symptoms with high coefficients in the cost function. The results can be applied in situations where a clinician wants to emphasize one symptom (i.e., major symptom) over the other symptom (i.e., minor symptom).

When different coefficients were applied to the objective functions of the TN and each objective function is quadratic, the experimental results suggest that the higher the relative coefficient value, the better the result for the corresponding symptom. This can be applied in situations where a clinician needs to manipulate the tradeoff among symptoms. This allows the clinician to adjust the dose according to the health status in each patient.

The simulation results of training the TN with a quadratic cost function and a quartic cost function indicate the threshold-like behavior in the quartic cost function when the dose is in the neighborhood of a "threshold" d^*. By adjusting the threshold and exponent, a bound can be imposed on a given symptom value.

The performances of the Quadratic Response Surface (QRS) and the networks (PM and TN) were compared in the third study. The shapes of surface plots generated by the PM are much closer to that of the ideal function than the QRS. In general, the performance of the networks is better than the QRS. However, the networks have some problems with local minima.

Computer simulations using distal learning techniques have been used to analyze optimal treatment outcomes for different dose adjustment combinations. Generally, it is not possible to completely understand the activity of drugs in the human body. Moreover, patients are already in a vulnerable situation. Receiving an excessive dose may result in fatal toxicity in patients. The merit of this method is that once the network is trained, it can be a useful consulting tool for a clinician in trying different combinations of doses to speculate the outcomes before prescribing them to the patients.

Future Research Directions

To model the invariability that simulates aspects of real data, and to investigate the robustness of the system, noise should be introduced in all studies. Nevertheless, the ultimate test of the framework is to use real medical data. Real world data may be noisy, incomplete, and have high dimensionality (i.e., demographic, clinical and laboratory variables). For safety reasons, however, preclinical studies using animals should be performed before beginning human trials. Animal studies allow for more explorations of dose space, which is important for training the PM.

Medical data may consist of multiple observations of patients over time. The change in symptoms over time is not captured in the current model. Utilizing information over time might result in a better understanding of a patient's prognosis than using isolated findings. To handle prognosis (predicted symptoms) of each patient at different intervals in time, the intuitive approach is to train the network at time *t*(*n*) with the subsequent time *t*(*n+1*) as the target of the outputs. This might be feasible using a simple recurrent network architecture (Elman, 1990). Use of time-oriented information is expected to enhance the performance of the network.

BIBLIOGRAPHY

Albert, D.A., Munson, R. and Resnik, M.D. Reasoning in medicine: An introduction to clinical inference. *The Johns Hopkins University Press. Baltimore and London. (1988)*

Agatonovic-Kustin, S., and Beresford, R., Basic concepts of artificial neural networks (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis: Vol. 22. pp.717-727 (2000).*

Agnieszka Onisko, Marek J. Druzdzel and Hanna Wasyluk. A Bayesian network model for diagnosis of liver disorders. *In Proceedings of the Eleventh Conference on Biocybernetics and Biomedical Engineering, pp. 842-846, Warsaw, Poland (1999).*

Armoni, A., Use of neural networks in medical diagnosis. *M.D. Computing; Vol. 15(2) pp:100-104 (1998).*

Bai, O, Nakamura, M. Nishida, S et.al. Markov Process amplitude EEG model for spontaneous background activity. *Journal of Clinical Neurophysiology. 18(3). Pp. 283-290 (2001).*

Barbolosi, D. and Illiadis, A. Optimizing Drug Regimens in Cancer Chemotherapy by an Efficacy-Toxicity Mathematical Model. *Computer and Biomedical Research 33, pp. 211-226 (2000).*

Baxt, WG. Use of an artificial neural network for the diagnosis of myocardial infarction. *Annals of Internal Medicine.115.pp:843-848. (1991).*

Baxt WG. Skora J. Prospective Validation of Artificial Neural Network trained to Identify Acute Myocardial Infarction. *Lancet 347, pp: 12-15 (1996).*

Behera, L, Gopal, M and Chaudhury, S "On adaptive trajectory tracking of a robot manipulator using iversion of its neural emulator" IEEE Transaction in Neural Network. Vol.7 no.6 Nov. (1996).

Boldrini, J.L. and Costa, M.I.S. Therapy Burden, drug resistance, and optimal treatment regimen for cancer chemotherapy. *IMA Journal of Mathematics Applied in Medicine and Biology. 17, pp.33-51. (2000)*

Bota A. Gella, F.J. and Canalias, F. Optimization of adenosine deaminase assay by response surface methodology. *Clinica Chimica Acta. 290(2): 145-57, Jan 5 (2000)*

Box G.E.P and Draper, N.R. Empirical Model-Building and Response Surfaces. *John Wiley & Son. (1987)*

Buchanan, B., Sutherland, G., and Feigenbaum, E. Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry. Machine Intelligence*, New York: American Elsevier. Vol. 4.(1976).*

Burnside, E, Rubin, D and Shachter, R. A Bayesian Network for mammography. *Proceedings/AMIA Annual Symposium, pp. 106-110 (2000).*

Cichocki, A and Unbehauen, R Neural Networks for Optimization and Signal Processing. *John Wiley & Sons, Inc (1993).*

Cho, P.S, Lee, S, MarkII, R.J. et.al. Optimization of intensity modulated beams with volumes constraints using two methods: Cost function minimization and projections onto convex sets. *Med.Phys. 25(4) April, 1998. pp.435-443. (1998).*

Chong, E.K.P and Zak, S.H. "An Introduction to Optimization" 2nd edition John Wiley & Sons, Inc. (2001).

Coiera, E., Artificial Intelligence in Medicine. *Guide to Medical Infomatics, the Internet and Telemedicine, Chapman and Hall Medical. (1997).*

Coiera, E., Artificial Intelligence in Medicine. http://www.coiera.com

Covell, D.G., Uman, G.C., and Manning, P.R. Information needs in office practice: are they being met? *Ann Intern Med 103. pp:596-9.(1985).*

Craig, J.J. Introduction to Robotics. *Addison-Wesley, Reading, M.A, (1986).*

Davis,D.T, Hwang, J.N and Lee, J.S. Improved network inversion technique for query learning: Application to automated cytology screening Proc. *Computer-Based Medical System. 4th Annual IEEE Symposium. Baltimore, MD. pp. 313-320 (1991).*

Del Vecchio, R.J. Understanding Design of Experiments: A Primer for Technologists. *Hander/Gardner Publications, Inc., Cincinnati. (1997)*

Demers, D and Kreutz-Delgado, K. Learning global direct inverse kinematics. *Advances in Neural Information Processing Systems 4. J.E.Moody, S.J.Hanson amd R.P.Lippmann. editors. Morgan Kaufmann, pp. 589-591 (1992).*

Derendorf, H, Lesko, L.J, Chaikin, P et.al. Pharmacokinetic/Pharmacodynamic Modeling in Drug Research and Development. *J.Clin. Pharmacol, 40:1399-1418 (2000).*

Drew, PJ., and Monson, JRT., Artificial intelligence for clinicians. *The Royal Society of Medicine; Vol. 92(3) March. pp:108-109. (1999).*

Dybowski, R., Gant, V., Artificial neural networks in pathology and medical laboratories. *Lancet; Vol. 346, pp.1203-1207. (1995).*

Eberhart, R.C. and Dobbins, R.W. Designing neural network explanation facilities using genetic algorithms. *Proc, Int. Joint Conference. Neural Networks, Vol. II, Singapore, pp. 1758-1763 (1991).*

Ellacott, S and Bose, D Neural networks: Deterministic Methods of Analysis. *An International Thomson Publishing Company (1996).*

Elman, J.L. Finding Structure in Time. *Cognitive Science*, *14, pp: 179-211 (1990).*

Elstein, A.S., Shulman, L.S., and Sprafka, S.A. Medical Problem Solving: An analysis of clinical reasoning. *Cambridge, M.A. Harvard University Press. (1978)*

Fassihi, R. Fabian J., and Sakr A.M. Application of response surface methodology to design optimization in formulation of a typical controlled release system. *Pharmazeutische Industrie. 57(12): 1039-1043 Dec (1995)*

Fieschi, M. Artificial Intelligence in Medicine Expert Systems *(translated by Cramp, D) Chapman and Hall. (1990)*

Fonner, D.E. Jr., Buck, J.R, and Banker, G.S. Mathematical Optimization Techniques in Drug Product and Process Analysis. *Journal of Pharmaceutical Science. Vol.59, No. 11, Nov (1970)*

Forsstrom, JJ., and Dalton KJ., Artificial neural networks for decision support in clinical medicine. *Annals of Medicine; Vol. 27(5), pp.509-517.(1995).*

Forsythe, D.E., Buchana, B.G., Osheroff, J.A., and Miller, R.A. Expanding the concept of medical information: an observational study of physicians' information needs. *Comput Biomed Res. 25.pp:181-200. (1992)*

Fu, K.S. Syntactic Pattern Recognition with Applications*, Prentice Hall, Eaglewood Cliffs, NJ. (1982a).*

Goolsby, M.J. Urinary tract infection. *J Am Acad Nurse Pract, Sep 13(9) pp. 395-398 (2001).*

Gullapalli, V Reinforcement Learning and Its Application to Control. *Dissertation Thesis. Department of Computer and Information Science. University of Massachusetts. Feb. (1992).*

Guihenneuc-Jouyaux, C, Richardson, S. and Longini, I.M Jr. Modeling markers of disease progression by a hidden Markov Process application to characterizing CD4 cell decline. *Biometrics. 56(3), pp. 733-741 (2000).*

Harry, E.P.Jr. Heuristic Methods for Imposing Structure on Ill-Structured Problems: The Structuring of Medical Diagnosis. *Artificial Intelligence in Medicine. AAAS Selected Symposium, pp.119-190 (1982).*

Herts, J. Krogh, A., and Palmer, RG., Introduction to the theory of neural computation. *Lecture Notes Vol.1. Persus Books, Reading, MA (1991).*

Higgs, J., and Jones, M. (editior). Clinical reasoning in the health professions. *Butterworth Heinemann. 2nd edition (2000).*

Hillier, F.S. and Lieberman Introduction Operations Research 5th edition. *The McGraw-Hill Publishing Company (1990)*

Hirshberg, A., and Adar, R., Artificial neural networks in medicine. *Israel Journal of Medical Sciences. Vol. 33(10), pp: 700-702 (1997).*

Holland, J.H. Adaption in natural and artificial systems. *University of Michigan Press, Ann Arbor. (1975)*

Hornik, K., Stinchcanbe, M., and White, H., Multilayer feedforward networks are universal approximators, Neural Networks*; Vol .2 (5): pp.359-366 (1989).*

Hoskins, D.A, Hwang, J.N. and Vagners,J Iterative inversion of neural networks and its application to adaptive control. *IEEE Transaction in Neural Network, Vol .3 Mar, pp. 292-301(1992).*

Iliadis, A and Barbolisi, D. Optimizing Drug Regimens in Cancer Chemotherapy by an Efficacy-Toxicity. *Mathematical Model Computers and Biomedical Research. 33, 211-226 (2000).*

Iliffe S, Austin T, Wilcock J, et.al. Design and implementation of a computer decision support system for the diagnosis and management of dementia syndromes in primary care. *Methods Inf Med, 41(2), pp. 98-104 (2002).*

Ishida, M and Zhan, J. Neural model-predictive control of distributed parameter crystal growth process. *AICHE J. Vol 41, No 10, Nov. pp. 2333-2336 (1995).*

Jasen, C.A, Reed, R.D. et. al. Inversion of Feedforward Neural Networks: Algorithms and Applications. *Proceedings of the IEEE. Vol.87. No 9. Sep. pp. 1536-1549 (1999).*

Jasen,C.A, Reed R.D.,et.al. Location of operating points on the dynamic security border using constrained neural network inversion. *Proc. Int. Conf. Intelligent System Applications to Power System, (ISAP'97), Seoul, Korea, July(1997).*

Jones, D., Neural network for medical diagnosis. *Handbook of Neural Computing Applications, Academic Press, pp: 309-318 (1990).*

Jordan, M.I. Attractor dynamics and parallelism in a connectionist sequential machine. *Proceeding of the Eighth Annual Conference of the Cognitive Science Society, pp 531-546, Amherst, MA (1986).*

Jordan, M.I. and Rumelhart, D.E. Forward Models: Supervised learning with a distal teacher. *Cognitive Science, 16, pp. 307-354. (1992).*

Jordan, M.I. Computational Aspects of Motor Control and Motor Learning. *Handbook of Perception and Action: Motor Skills. New York: Academic Press, H.Heuer and S.Keele. editor(1996).*

Kahn, C.E, Laur, J.J and Carrera, G.F. A Bayesian Network for diagnosis of primary bone tumors. *Journal of Digital Imaging. 14(2 Suppl 1), pp 56-57 (2001).*

Karmarkar, N. A new polynomial time algorithm for linear programming. *Combinatorica 4. pp. 373-395 (1984).*

Kennedy, R.L., Harrison, R.F., Burton, A.M., et.al. An artificial neural network system for diagnosis of acute myocardial infarction (AMI) in the accident and emergency department: evaluation and comparison with serum myoglobin measurements. *Artificial Intelligent in Medicine. 52, pp:93-103. (1997)*

Kleinbaum, D.G,Kupper, L.L and Muller, K.E. Applied Regression Analysis and Other Multivariable Methods. The University of North Carolina at Chapel Hill. PWS-KENT Publishing Company (*1998*).

Khuri, A.I. and Cornell, J.A. Response Surfaces: Designs and Analyses. *Marcel Dekker, Inc. (1987)*

Krishan, A., Sridhar, K.S. Mou, C. et.al. Synergistic effect of Prochlorperazine and Dipyridamole on the cellular Retention and Cytotoxicity of Doxorubicin. *Clinical Cancer Research. Vol. 6, pp 1508-1517. (2000).*

Langer, M and Leong J. Optimization of beam weights under dose-volume restrictions. *Int. J.Radiat.Oncol.Biol.Phys.13,1255-60 (1987)*

Lavrac, N. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine, 16:pp.3-23. (1999).*

Lisboa, PJG., Vellido, A., and Wong, H., Outstanding issues for clinical decision support with neural networks. *The Proceedings of the ANNIMAB-1 Conference Series Perspectives in Neural Computing (Springer-Verlag) (1999).*

Leondes, C.T. editor Optimization Techniques. *Volume 2 of Neural Network Systems Techniques and Applications. Academic Press (1998).*

Linden, A and Kinderman, J. Inversion of multiplayer nets. *Proc. Int. Joint Cond. Neural Networks, Vol. II, Washington D.C., pp.425-430 (1989).*

Looney, CG., Pattern recognition using neural networks: theory and algorithms for engnineers and scienti[47]sts. *Oxford University Press, (1997).*

Lu, B.L. and Ito, K. Regularization of Inverse Kinematics for Redundant Manipulators Using Neural Network Inversions. *Proc. IEEE. Int. Conference Neural Networks. Perth. Australia. Nov. 27-Dec1, pp 2726-2731(1995).*

Lu, B.L, Kita, H and Nishikawa, Y. Inverting Feedforward Neural Networks Using Linear and Nonlinear Programming. *IEEE Transactions on Neural Networks. Vol. 10, No.6. Nov, pp. 1271-1290 (1999).*

Lucas, PJF., and Abu-Hanna, Ameen. Prognostic methods in medicine. *Artificial Intelligence in Medicine. 15:pp:105-119 (1999).*

Lucila, OM., and Rowland T., Neural network applications in physical medicine and rehabilitation. *American Journal of Physical Medicine & Rehabilitation;Vol. 78(4) July/Aug . pp: 392-398.(1999).*

Mango, LJ., Reducing false negatives in clinical practice: the role of neural network technology. *American Journal of Obstetrics and Gynecology; Vol. 175(4), Oct. pp. 1114-1119. (1996).*

Mango, LJ., Computer-assisted cervical cancer screening using neural networks. *Cancer letter; Vol. 77, pp: 155-162. (1994).*

Mateev, A and Savkin, A.V. Optimal chemotherapy regimens: influence of tumors on normal cells and several toxicity constraints. *IMA Journal of Mathematics Applied in Medicine and Biology. 18. 25-40. (2001).*

Meibohm, B and Derendorf, H. Minireview: Pharmacokinetic/Pharmacodynamic Studies in Drug Product Development. *Journal of Pharmaceutical Sciences 91(1) , pp: 18-30(2002).*

Mehrata, K., Mohan, C.K., and Kanka, S., Elements of artificial neural networks.*MIT press, Cambridge, MA. (1997).*

Miller, R.A., Pople, H.E., and Myers, J.D. INTERNIST-1: An experimental computer-based diagnostic consultant for general internal medicine*, New England Journal Medicine, 307. pp.468-476.(1982).*

Miller, R.A. Medical Diagnosis Decision Support Systems – Past, Present and Future. *American Journal of Medical Informatics Associations. 1,pp.8-28. (1994).*

Mohan R and Wand X-H. Physical vs Biological Objectives for Treatment plan optimization. *Radiother.Onco. 40,186-7 (1996)*

Musen, M.A., Shahar, Y. and Shortliffe, E.H. Clinical decision support systems. Computer Applications in Health Care and Biomedicine, Second Edition. *(eds: Shortliffe, E.H, Perreault, L.E, Wiederhold, G. and Fagan, L.M) Springer-Verlag. (2000).*

Mulsant, BH., A neural network as an approach to clinical diagnosis. *M.D. Computing Journal; Vol. 7(1). pp.25-36 (1990).*

Munro, P.W. Dual backpropagation: a scheme for self-supervised learning. *Proceedings of the 9th meeting of the Cognitive Science Society. Princeton, N.J.:Erlbaum (1987).*

Munro, P.W. and Sangauansintukul, S. A Neural Network Approach to Treatment Optimization. *To be appeared in AMIA, November 9-13 (2002).*

Munro, P.W. and Sanguansintukul, S. Treatment Optimizaton with a Neural Control System. *To be appeared ICONIP'02, Singapore. November 18-22 (2002).*

Myers, R.H. and Montgomery, D.C. Response Surface Methodology – Process and Product Optimization Using Designed Experiment*, John Wiley & Sons. (2002)*

Naguib, RNG., Sherbert, GV., Artificial neural networks in cancer research. *Pathbiology; Vol 67, pp129-139.(1997).*

Nash, S.G and Sofer, A. Linear and Nonlinear Programming. *The McGraw-Hill Publishing Company (1996)*

Niemierko, A. A Selection of Objective Functions for Optimization of Intensity Modulated Beams. *Med.Phys.23,1172.(1996)*

Osheroff, JA., Forsythe, DE., Buchanan, BG, Bankowitz, RA, Bluemfeld BH, and Miller, RA. Physicians information needs: an analysis of questions posed during clinical teaching in internal medicine. *Ann Intern Med. 114. pp:576-581. (1991).*

Owens D.K and Sox H.C. Medical Decision Making: Probabilistic Medical Reasoning in Medical Informatics. Computer Applications in Health Care and Biomedicine, Second Edition. *(eds: Shortliffe, E.H, Perreault, L.E, Wiederhold, G. and Fagan, L.M) Springer-Verlag. (2000).*

Pacific Northwest National Laboratory's Environmental Molecular Sciences Laboratory World Wide Web site (http://www.emsl.pnl.gov:2080/proj/neuron/).(2000).

Pauker, S.G., Gorry, G.A. Schwartz, W.B. and Kassirer, J.P. Towards the simulation of clinical cognitions. *American Journal of Medicine. 60, pp.981-996 (1976).*

Pauker, S.G. and Szolovits, P. Analyzing and Simulating Taking the History of the Present Illness: Context Formation*, in Computational Linguistics in Medicine, (eds Schneider and Sagwall-Hein), North-Holland, Amsterdam. (1977).*

Penny, W., and Forst, D., Neural networks in clinical medicine. *Medical Decision Making; Vol. 16(4), pp.386-398 (1996).*

Pople, H.A.Jr., Myers, J.D. and Miller. R.A. DIALOG: A Model of Diagnosis Logic for Internal Medicine*, Proceedings of 4th I.J.C.A.I., MIT Press Artificial Intelligence Lab., Cambridge, Massachusettes.(1975).*

Reed, RD., and Mark II, RJ., Neural smithing: supervised learning in feedforward artificial neural networks. *The MIT press, Cambridge, MA. (1998).*

Reggia, JA., Neural computation in medicine. *Artificial Intelligence in Medicine; Vol.5, pp.143-157(1993).*

Rognvaldsson, T. On Langevin updating in multiplayer perceptrons. *Neural Computing. 6, pp:916-926 (1994).*

Rumelhart, D., Hilton, G., and William, R., Learning internal representation by error propagation appeared in parallel distributed processing. *Edited by Rumelhart and Mc.Clelland, MIT Press, (1986).*

Rustagi, J. Optimization Techniques in Statistics. *Academic Press (1994).*

Sabbatini, RME., Applications of connectionist systems in biomedicine. *Proceedings of the 7th International Congress in Medical Informatics, Geneva, Switzerland. Sep, Amsterdam: North Holland, pp : 418-425 (1992).*

Sahiner, B, Chan, HP, Petrick, N et.al. Design of a high-sensitivity classifier based on a genetic algorithm application to computer-aided diagnosis. *Physics in Medicine & Biology 43(10), pp 2853-2871 (1998).*

Schwartz, JB, Flamholz, J.R. and Press, R.H. Computer Optimization of Pharmaceutical Formulations I: General Procedure. *Journal of Pharmaceutical Science, Vol.62, No.7, July (1973)*

Shortliffe, E.H and Buchanan, B.G. A Model of inexact reasoning in medicine. *Math Biosci,23, pp.351-379 (1975).*

Shortliffe, E.H. Computer-Based Medical Consultations – MYCIN. *New York: Elsevier/North Holland (1976).*

Shortliffe, EH., The adolescence of AI in medicine: will the field come of age in the '90s*? Artificial Intelligence in Medicine; Vol. 5, pp.93-106 (1993).*

Shortliffe, E.H., and Barnett, G.O. Medical data: their acquisition, storage, and use. *Medical Informatics. Computer Applications in Health Care and Biomedicine. Springer-Verlag. Second*

Edition. (eds: Shortliffe, E.H, Perreault, L.E, Wiederhold, G. and Fagan, L.M) Springer-Verlag. (2000).

Shyy, W.,Tucker, P.K. Vaidyanathn, R., Response Surface and Neural Network Techniques for Rocket Engine Injector Optimization. *35th AIAA/ASME/SAE/ASEE Joint propulsion Conference and Exhibit, June 20-24, AIAA-99-2455, Los Angeles, CA (1999).*

Stengel, R.F., Ghigliazza, R., Kulkarni, N and Laplace, O. Optimal Control of Innate Immune Response. *Optimal Control applications and methods. 23:91-104. (2002)*

Stewart, W.H. Application of response surface methodology and factorial designs to clinical trials for drug combination development. *Journal of Biopharmaceutical Statistics. 6(3): 219-30. Jul (1996)*

Swan. G.W. Application of Optimal Control Theory in Biomedicine. *Dekker, New York, (1984).*

Swan, G.W. Role of Optimal Control Theory in Cancer Chemotherapy. *Math.Biosci.101,237 (1990).*

Szolovits, P. Artificial Intelligence in Medicine. *AAAA Selected Symposium Series . Westview Press, Inc.(1982).*

Szolovits, P. Artificial Intelligence and Medicine. *Chapter 1 in Szolovits, P. (Ed.) Artificial Intelligence in Medicine. Westview Press, Boulder, Colorado (1982).*

Tafeit, E., and Reibnegger, Gilbert., Artificial neural networks in laboratory medicine and medical outcome prediction. *Clin Chem Lab Med;Vol. 37(9), pp. 845-853 (1999).*

Tewari, A., Artificial intelligent and neural networks: concept, applications and future urology. *British Journal of Urology;Vol. 80(supp. 3) Nov. pp53-58. (1997).*

Tversky, A., and Kahneman, D., Judgment under uncertainty: heuristics and biases. *Science. 185:1124-1131. (1974)*

Vinterbo, S and Ohno-Machado, L. A genetic algorithm approach to multi-disorder diagnosis. *Artificial Intelligence in Medicine. 18(2), pp. 117-132 (2000).*

Vinterbo, S and Ohno-Machado, L. A genetic algorithm to select variables in logistic regression: example in the domain of myocardial infarction. *Proceedings/AMIA Annual Symposium, pp. 984-989 (1999).*

Weiner, M.G., and Pifer, E. Computerized decision support and the quality of care. *(Information Technology) Managed Care. May, pp: 41-51. (2000).*

Weiss, S.M., Kulikowski, C.A., Amarel, S. and Safir, A. A model-based method for computer-aided medical decision making. *Artificial Intelligence. 11, pp.145-172, 1978.*

WebMD (http://www.webmd.org)

Werbos, PJ., The roots of back-propagation*, Wiley, New York (1994).*

William, GB., Application of artificial neural networks to clinical medicine. *The Lancet;Vol. 346(8983) Oct. pp.1135-1138.(1995).*

Williams, R.J. Inverting a connectionist network mapping by backpropagation of error. *Proc. 8th Annual. Conference. Cognitive Science Society. Hillsdale. N.J: Lawrence Erlbaum, pp. 859-865(1986).*