OPTIMIZATION DECISION MAKER ALGORITHM FOR INFRASTRUCTURE INTERDEPENDENCIES WITH I2SIM APPLICATIONS

by

Ming Bai

B.A.Sc., The University of British Columbia, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

July 2012

Abstract

The study of complex interdependent systems is an important research area. In recent years, it has been applied to disaster response management and building energy systems. I2Sim (Infrastructures Interdependencies Simulator) is a software simulation toolbox developed by the Power Lab at the University of BC. It has a wide range of capabilities including simulation of disasters scenarios and energy system optimization. The user needs to provide Human Readable Tables (HRTs) as inputs for the program. The basic ontology of the I2Sim Resource Layer includes cells, channels and tokens, which are abstractions from real life objects.

Initially, the intent of this thesis was to examine the energy usage pattern of the Kaiser Building, perform energy optimization modeling and examine how it relates to energy policies. After some initial research, it was not possible to proceed further due to a lack of metered data.

The research focus was changed to disaster scenario simulation. This thesis proposes a new optimization algorithm named Lagrange Based Optimization (LBO). The main objective is to maximize the number of discharged patients from the hospitals simulated in this study. The first scenario modeled is a three-hospital scenario with no transportation to illustrate the principles of the algorithm. Then a three-venue three-hospital scenario with transportation was modeled to maximize both the number of patients transported to the hospitals and the number of patients discharged from the hospitals. After that, the first scenario is compared against the performance of a Reinforcement Learning (RL) agent method concurrently developed in the same research group. Overall, the LBO algorithm demonstrates optimal results in the various I2Sim modeling scenarios.

Preface

Parts of this thesis appeared in the ISCRAM 2012 Conference. Dr. Kui Wang, Dr. K.D. Srivastava, Dr. Jose Marti and I co-authored a paper titled "Optimal Decision Maker Algorithm for Disaster Response Management with I2Sim Applications". It was accepted and published in the conference proceedings [1]. The full reference is as follows.

[1] K. Wang, M. Bai, K.D. Srivastava and J. Marti. "Optimal Decision Maker Algorithm for Disaster Response Management with I2Sim Applications," *ISCRAM Conference*, 2012, pp. 1-5.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AGC – Automatic Generator Control

AI – Artificial Intelligence

ANN – Artificial Neural Networks

DAARTS - Decision Assistant Agent in Real Time Simulation

DRNEP – Disaster Response Network Enabled Platform

EDC – Economic Dispatch Control

EOC – Emergency Operations Centre

GWh – Gigawatt Hour

HRT – Human Readable Table

HVAC – Heating Ventilation and Air Conditioning

I2Sim – Infrastructures Interdependencies Simulator

kL – Kiloliter

kW – Kilowatt

LBO – Lagrange Based Optimization

LUT – Look Up Table

MCI – Mass Casualty Incident

MW – Megawatt

OVNI – Object Virtual Network Investigator

PM – Physical Mode

RL – Reinforcement Learning

RM – Resource Mode

SCADA – Supervisory Control and Data Acquisition

Acknowledgements

First of all, I would like to thank my co-supervisors Dr. K.D. Srivastava and Dr. Jose Marti. They have been very patient and helpful throughout my time at the Power Lab. I am grateful to Dr. Hermann Dommel for being on my examining committee. I would like to greatly thank Dr. Kui Wang for the technical guidance she has provided throughout my Master's studies.

This work would not have been possible without the help and collaboration from the many Power Lab friends and colleagues. In particular, I would like to thank Mohammed Khouj for helping me learn his Reinforcement Learning agent method. I also like to thank Justin Wang, Cesar Lopez and Paul Lusina for their help in preparation of my thesis.

I would like to thank Modern Green Development Ltd. and UBC Sustainability Initiative for providing an once-in-a-lifetime internship opportunity. The internship enabled me to work in Modern Green's headquarter in Beijing for two months, and I researched into ways of optimizing building energy usage.

Last but not least, I would like to thank my parents and Mr. John Vandermaar for their tireless help and support throughout the years. I would not be where I am without them.

Dedicated to My Parents

1 Introduction

The subject of this thesis is the development of an optimization algorithm for resource allocation named Lagrange Based Optimization (LBO), to be used in the Infrastructure Interdependencies Simulator (I2Sim) toolbox. As a validation of its performance, LBO is compared against a Reinforcement Learning (RL) method. Both methods are evaluated with modeling scenarios built with the Infrastructures Interdependencies Simulator (I2Sim) toolbox, a software package developed at the Power Lab at UBC. The main objective of the modeling scenarios is to optimally distribute the limited resources, and enable the hospitals to discharge the maximum number of patients. Overall, both methods have their advantages and disadvantages. They complement each other and may be integrated into the future implementation of the I2Sim Decision Layer, which is still in conceptualization.

1.1 Motivation for the Research

Many large scale disasters have occurred in recent years. The most notable ones include the Triple Disasters in Japan of 2011 (earthquake, tsunami and nuclear power plant meltdown), the Richter 8.0 Earthquake in China of 2008, and Hurricane Katrina which destroyed parts of New Orleans in 2005 [1]. These disasters caused terrible loss of lives, and major impacts to the economies of the nations affected. After a disaster occurs, the rescue of human lives becomes a top priority. With financial support from the Government of Canada, the I2Sim group at UBC's Power Lab started research into critical infrastructures interdependencies and disaster scenario modeling [2]. The past works in disaster scenarios included I2Sim models that simulated an imagined earthquake in Downtown Vancouver during the 2010 Winter Olympics [3], re-enacted a historical power outage and its restoration on the UBC Campus [4], and mimicked the Japan

Sendai tsunami and its subsequent evacuations [5]. Recently, I2Sim has been applied to energy modeling of the UBC Living Lab project. One thesis has developed an I2Sim multi-energy simulator for the management of Greenhouse Gases (GHG) emissions and costs [6]. Another thesis has built an I2Sim financial production cell for the calculations of purchasing resources [7]. There has not been any in-depth examination of resource allocations and decision support. By making informed decisions based on known characteristics of a system of systems and the available amount of resources, the amount of damage after a disaster occurs can be minimized. The Lagrange Based Optimization (LBO) algorithm proposed here is not meant to replace human emergency responders or compete with any other methods. Instead it may be added as a training tool to help emergency responders learn and to improve their decisions.

1.2 Challenges and Research Needs

To make an informed decision on resource allocation, it is important to obtain a thorough understanding of the environment to be modeled, which is a complex interdependent system with many critical infrastructures. According to a report by the Idaho National Laboratory [8], critical infrastructure interdependency modeling has challenges that are similar to other modeling areas. Examples of the challenges include how readily accessible are the data, how easy it is to develop a working model, and how easy it is to validate the model. According to Rinaldi et al. [9], the development of an encompassing framework for the modeling of various infrastructure interdependencies is a major obstacle. The Idaho National Laboratory report agrees with this challenge by mentioning that interdependency modeling requires an extremely large number of cross sector examinations [8]. As a simple example, a water pump station is reliant on input from an electrical substation to operate and deliver the water to where it is needed. In a complex environment, the decision making process needs to consider the infrastructures' internal characteristics, the types of failures and the states of operation [9].

1.3 Research Objectives

Initially, the research objective of this thesis was to be an energy simulation of the Kaiser Building as a complex interdependent system. I examined a number of software packages on the market and compared their advantages and disadvantages. The detailed information from the Kaiser Building research can then be coupled with the results from the Living Lab project, which examined energy usage of the entire UBC Vancouver campus. After a while, it was determined that the Kaiser Building was not separately metered by UBC Utilities. It became difficult to proceed with this building energy simulation project and it had to be abandoned. The preliminary research work performed on this topic will be presented in Chapter 2.

After the abandonment of the Kaiser Building project, the research continued into the general area of complex interdependent systems. The decision making process is an interesting and very vital part of a complex interdependent system and it was selected as the topic. The primary research objective of this thesis is to evaluate the suitability of the proposed Lagrange Based Optimization (LBO) algorithm with I2Sim. The secondary research objectives include: 1) compare the LBO algorithm with the performance of a Reinforcement Learning (RL) agent method, and 2) examine the advantages and disadvantages of both methods.

As mentioned before, the I2Sim software package has been successfully employed to model disaster scenarios before, and it was chosen as the main software simulation tool for this thesis.

This work examines how the decision maker algorithms could be applied to I2Sim modelling scenarios. As mentioned before, the main objective is to maximize the number of patients discharged from the hospitals. The development of the LBO algorithm may be very useful for future research work, particularly the implementation of the I2Sim Decision Layer.

1.4 Organization of the Thesis

This thesis is organized into the following chapters.

- The first chapter introduces the motivation for the research, the research needs and the challenges and the research objectives.
- The second chapter presents the background information on the I2Sim Toolbox, complex interdependent systems, the Reinforcement Learning (RL) method and the preliminary research work performed on the Kaiser Building.
- The third chapter discusses optimization in general, the Lagrange Multiplier method in continuous domain and the algorithm of the Lagrange Based Optimization (LBO) method.
- The fourth chapter focuses on the I2Sim simulation scenarios. There are two I2Sim modeling scenarios created. The first is three hospitals with no transportation, meaning patients are already inside the hospitals' waiting rooms. The second scenario is three venues and three hospitals with transportation, meaning patients need to be transported from the venues to the hospitals. The scenario of the three hospitals with no transportation is used to evaluate both the LBO and RL methods. The three venues and three hospitals scenario is only used to evaluate the LBO method.
- The fifth chapter summarizes the main conclusions of this thesis and discusses the future works to be performed.

2 Background Information and the Kaiser Building Research

This chapter discusses the background information on complex interdependent systems, the I2Sim toolbox and the Kaiser Building research. In recent years, governments are allocating more resources into the research area of infrastructures interdependencies in complex interdependent systems. To understand how infrastructures interdependencies work is important not only to prepare for disaster responses, but it can also be used for the optimization of peacetime energy consumption. According to Zimmerman [10], in a system of systems (such as a city), the interdependencies among the critical infrastructures (power grids, roads, water pipes, etc.) are important points of vulnerability that can compromise the normal state of operation. This is especially of concern during extreme events [10]. If there is a good understanding of the various interdependencies, then optimizing the numerous decisions made after a disaster strikes can help minimize the loss of human lives and improve the repair of critical infrastructures [11].

2.1 Complex Interdependent Systems

A complex interdependent system represents the network of real world infrastructures that are bi-directionally dependent on each other to function normally. According to Bagheri and Ghorbani [12], critical infrastructures are "complex networks of adaptive socio-technical systems" and they provide the most essential services for a society to function properly. When one part of the system experiences a failure, the mutual interdependencies amongst the infrastructures may jeopardize the normal course of operations [12]. To identify the interdependencies, one may take a systematic method to identify the main physical components and identify the functions of each component [13]. An example of critical infrastructure interdependencies can be described as follows: an oil refinery supplies fuels and lubricants to the transportation sector, and in return the transportation sector ships the oil products to their destinations [14]. Figure 1 is an adapted image from the report by the Idaho National Laboratory [8]. It illustrates the infrastructure interdependencies that exist amongst the different sectors.

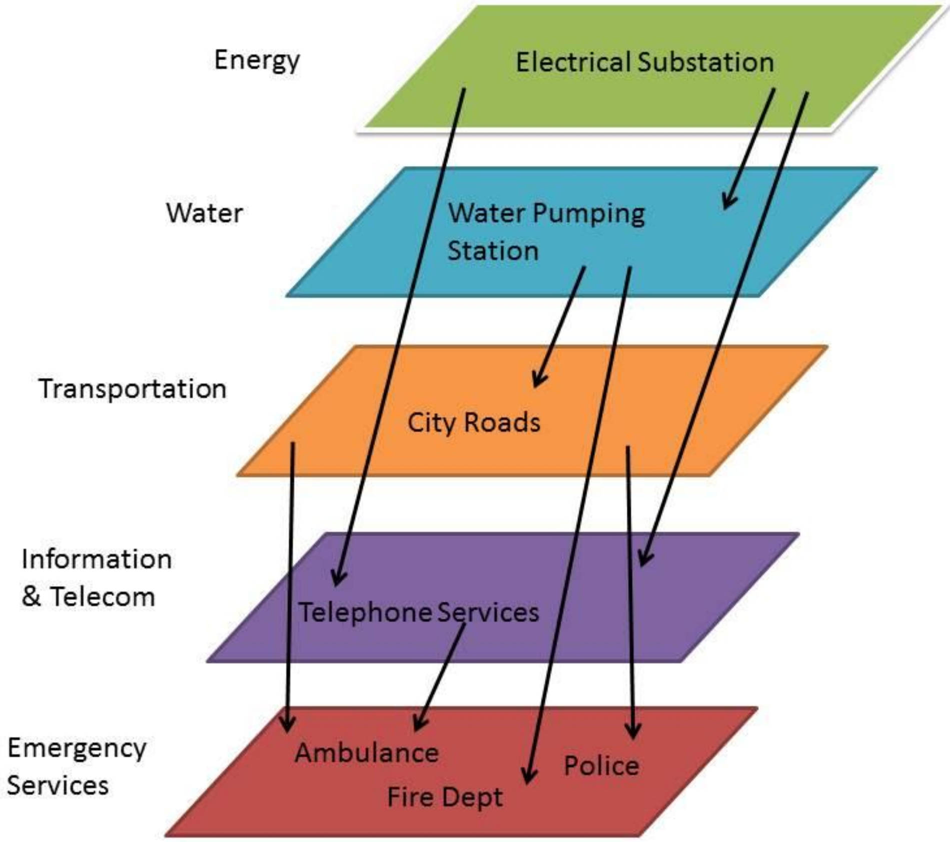

Figure 1: Infrastructure Interdependencies amongst Different Sectors [8]

In terms of the types of interdependencies, different sources have slightly varying definitions [8], [9]. Rinaldi et al. categorizes the interdependency types as follows [8]:

- Physical: two infrastructures rely on material outputs of each other to function
- Cyber: two infrastructures rely on informational outputs of each other to function
- Geographical: two infrastructures are both affected by a local environmental incident (e.g. an earthquake)

- Logical: two infrastructures are affected by each other by a mechanism that is neither of the three mentioned before (physical, cyber, geographical)

Pederson et al. classify interdependency types with some expansions based on the Rinaldi definitions [9].

- Physical: two infrastructures need to have physical links to be operational
- Informational: the communication links between two infrastructures need to be operational
- Geospatial: similar to geographical, an infrastructure is affected due to proximity to a disaster area
- Policy/Procedural: a cause-effect relationship to affect one infrastructure due to changes in another
- Societal: an infrastructure may be affected by psychological factors such as public perception, fear, etc.

2.2 I2Sim Toolbox

As mentioned previously, the I2Sim toolbox is the main simulation software used in this thesis. The toolbox was developed at UBC's Power Lab over a number of years, and new functions are continuously being added. While the initial motive to develop the I2Sim toolbox was to model disaster scenarios, the capabilities have expanded to include financial applications. The I2Sim toolbox is programmed with Matlab/Simulink. Its user interface has a drag-and-use design, which gives it the advantages of being simple to learn and use. The user does not need to know about the inner workings of the blocks [3], [15]. The toolbox is constituted of modular blocks that mimic real life entities such as buildings, pipes, electricity, etc.

There are some similarities of Petri net compared to I2Sim in terms of ontology [3]. A Petri net is a graph-based modeling tool capable of simulating infrastructure interdependencies [16]. Its main modeling components consist of transitions, places, tokens, and directed arcs. The transitions represent an event such as the disruption of electrical power. The places denote sites of production such as power plant and oil refinery. The tokens represent the resources being produced. The arcs connect places and transitions. Petri net uses the flow of tokens to show the state of the infrastructures and the interdependencies amongst them. Its main advantage is a simple visualization of the infrastructure interdependencies through the flow of the tokens [16]. When compared to I2Sim, Petri net lacks the ability to model quantitative information [3].

According to Marti et al. [17], I2Sim takes on a generalized systems engineering approach. The basic task is to transport resources from the point of origin to needed areas. The I2Sim simulator utilizes logical relationships between entities and quantities, expressed in mathematical equations. This allows the I2Sim to use mathematical tools and theories to solve complex network systems. The mathematical approach also allows further optimization in resource allocation [17]. For example, a hospital entity can be represented by various inputs and discharged patients as outputs.

2.2.1 I2Sim Ontology

The basic ontology of I2Sim's Resource Layer abstracts elements from real life and represents them in block form. This fundamental ontology is very general and can be applied to many types of modeling. The basic elements of I2Sim ontology consist of the following elements [3], [18].

- **Token:** A token is a unit that circulates throughout a model and can be inputs or outputs. Examples include water, electricity, ambulance and patients.
- **Cell:** A cell is a production unit where tokens are taken as inputs, transformed and produced as outputs. Examples include hospitals, electrical substations and water stations.
- **Channel:** A channel is a conduit where tokens are transported, but no transformation take place. Examples include roads, transmission lines and water pipes.
- **Control:** A control is a decision point where resources are allocated to different cells. Examples include water and electricity distributors.

Based on the above ontology, the I2Sim toolbox used in this thesis (Version 340) has the following components [7], [19]:

- **Aggregator:** A block that takes in multiple inputs and sums their total as the output.
- **Delay Channel:** A block that transports tokens from one point to another and delays them by a specified amount of time.
- **Distributor:** A block that takes one input and distributes it to several outputs according to specified distribution percentages.
- **Modifier Cell:** A block that applies weight factors to inputs and produces output according to a relationship.

- **Production Cell:** A block which can have one to several input(s), produces an output according to a pre-defined Human Readable Table (HRT). Examples include water and electrical substation and hospitals.
- **Source:** A block where tokens originate. Examples include water and electricity sources.
- **Storage:** A block where tokens can be stored and/or extracted. Example includes a hospital waiting room.
- **Control Panel:** A block that enables the user to specify the length of simulation, time step size and time units.
- **Visualization Panel:** A block that displays outputs taken at the probe.
- **Probe:** A block that monitors the signal output at a certain point and displays it.

The initial development of the I2Sim toolbox is based on the methodology for the Object Virtual Network Investigator (OVNI) simulator [4]. OVNI was also developed at the UBC Power Lab, and it is used for the simulation of large power systems. The components in I2Sim are identified and categorized into a system of matrices called the infrastructure matrix. The solution of the infrastructure matrix is then determined by running the program. To obtain the matrices' solutions, Human Readable Tables (HRTs) are required as inputs [4].

2.2.2 Human Readable Table (HRT)

At the core of the I2Sim production cells are the Human Readable Tables (HRTs). HRTs describe the relationship between input(s) and output at different discrete levels. A HRT is basically a look up table in which a user can search the level of output of a production cell given the corresponding combination of inputs. In I2Sim, the current version of HRTs can have one to five discrete rows or thresholds. It is meant be a simplified tool, and the numerical entries in a HRT should be rounded to integers whenever possible. These thresholds are designed for decision support and to limit the state space of the problem. The current version of the HRT is a significant improvement from a previous version constructed by Liu in the same research group [3]. In Liu's version, she tried to identify all of the possible combinations, with many inputs and outputs, which resulted in hundreds of rows [3], [4]. The drawback of this is that the size of the table increases exponentially as the number of inputs increase.

Other concepts that relate closely with the HRT concept are Physical Mode (PM) and Resource Mode (RM). In the production cells where the HRTs are stored, there are associated Physical Modes (PM) and Resource Modes (RM). A Physical Mode is the physical integrity of a production cell [3]. A Resource Mode is how much resources are being supplied to the production cell [3]. Each of the PM and RM are discretized into 5 levels or thresholds, with level 1 being the highest and level 5 being the lowest [3], [7]. As mentioned before, the five thresholds are designed to limit the state space of the modeling to a manageable size, for instance 100%, 75%, 50%, 25% and 0%. For the sake of computer simulation, it has the advantage of simple implementation. The PM of a production cell limits the number of RMs available. For example, if the PM is at level 1 (100%), then there are five RMs available to the cell. But if the PM is at level 5 (0%), there is only one RM available. In an I2Sim production unit, the colour of the upper left corner shows the Physical Mode. The colour of the rest of the block denotes the Resource Mode. The colour codes are: Green (level 1), Blue (level 2), Yellow (level 3), Orange (level 4) and Red (level 5) [7]. In addition, a production cell also has two output ports indicating the level of PM and RM, where the outputs are discrete integers from 1 to 5.

Under normal circumstances, both PM and RM are at the highest levels. But after an uncontrollable event such as an earthquake, one or both may be compromised. The PM of a building may be 1 (100%) after an earthquake, but if the RM is 3 (50%), then the building is only 50% operational. Figure 2 shows a sample HRT with 5 rows. The PM associated with this HRT is at level 1, and all five RMs are available. In order to reach one of the rows or thresholds in a HRT, all of the inputs need to be at least at that threshold level. Figure 2 shows an example HRT. In order for the output to reach row 3, input 1 needs to be 15, input 2 needs to be 20 and input 3 needs to be 50.

Row	Output	Input 1	Input 2	Input 3
1	10	20	50	90
2	7	18	30	70
3	5	15	20	50
4	2	10	10	30
5	0	0	0	0

Figure 2: An Example HRT

2.2.3 Future Developments of I2Sim

Future developments of I2Sim may include a Decision Layer. Currently the modeling is performed in the I2Sim Physical Layer. The Decision Layer may be visualized as another level above the Physical Layer that interacts with it and makes decisions. As well, I2Sim would become an integral part of the DR-NEP (Disaster Response Network Enabled Platform) system currently being developed at the UBC Power Lab. The Lagrange Based Optimization algorithm could become the core calculation method of the Decision Layer. The Physical Layer first sends information on the available amounts of resources to the Decision Layer. The Decision Layer runs the LBO algorithm, and obtains the optimal decisions. Then the Decision Layer sends the allocation decisions back to the Physical layer.

The ontology of the I2Sim Decision Layer includes the following components [20]:

- **Decisions:** A table that relates time to distribution of a particular I2Sim token.
- **Decision Points:** The interconnection between the Physical Layer of I2Sim and Decision Layer.
- **Ln-Decision Maker:** n represents the set of integers starting from 0. When n is 0 it represents the infrastructure operator, and the largest n represents the highest decision maker.
- **Physical Data:** The information from the Physical Layer that is used by the Decision Makers at the Infrastructure and L0 to the Emergency Operations Centre (EOC) level.
- **Operating Data:** The information operators exchange with each other to keep proper operation.
- **Rule Base:** A set of rules used to process data and generate decisions.
- **Policies:** A set of meta-rules used to determine the applicable rule base for particular decisions.

2.3 Review of Alternate Approaches to Resource Allocation

The literature search found many alternate approaches to disaster resource allocation. According to Li et al., emergency resource scheduling is a key component of Emergency Management Systems [21]. Their problem is to optimize the transport path. Based on some statistical uncertainty, they calculate how to minimize the path from point A to B.

Zhou and She presented two solutions to resource allocation [22]. The first solution concerns how to distribute resources from a single supply point to several disaster areas, based on an integer programming model. The second solution concerns how to distribute resources from several supply sources to one disaster area, with the solution method based on the Topsis method. Their objectives are to minimize transportations costs and transportation time. The multi supply points to single disaster point method makes many simplifying assumptions, including the supply demands at the disaster areas are known and unvarying and the communication links are intact. In reality these may not hold true in a real disaster situation.

In addition to considering how to allocate the resources, the granularity of the patients should also be considered. In this thesis the simplifying assumption is made that all of the patients are the same. According to Donner et al., efficient management of mass casualty incidents (MCI) is a complex task. Emergency resources must be switched from a normal mode of operation to a temporary "disaster mode" [23]. The triage (assignment according to level of injury) of patients on site becomes very important. Figure 3 shows a chain of supply where patients are processed from point of disaster to the hospital. Figure 4 shows communication channels in an MCI [23]. Both figures are used with permission from the authors.

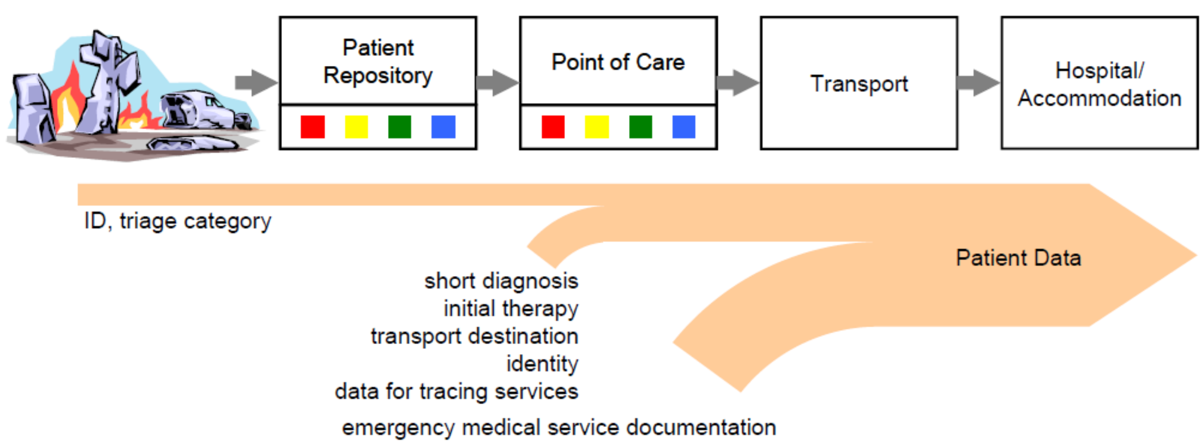

Figure 3: MCI Supply Chain Stations and Patient Data [23]

For future modeling of patient triage, there can be four broad categories: immediate, urgent, minor or deceased [23]. To ensure successful and optimized executions of the patient rescue and the transportation process, it is essential to obtain timely overviews of the patients and their triage categories.

Equation 2.1 provides an approximation to assess the seriousness of a disaster (S) [23]. In this equation, only three categories of triage are considered, the deceased patients are excluded.

$$S = \frac{number\ of\ T_1 + number\ of\ T_2}{number\ of\ T_3} \qquad (2.1)$$

S – Seriousness of the disaster

T_1 – Number of immediate patients

T_2 – Number of urgent patients

T_3 – Number of minor patients

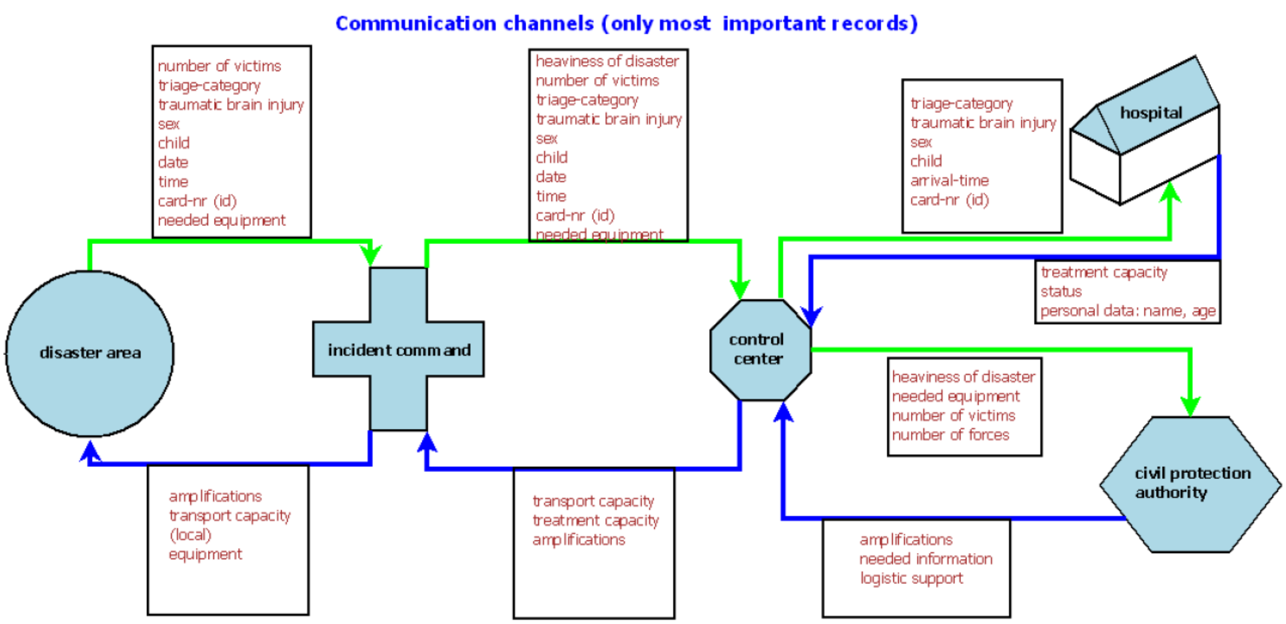

Figure 4: Communication Channels in an MCI [23]

After a disaster occurs, the transportation of the patients from disaster areas to hospitals becomes very important. Some of the questions to consider include: the number of patients, the priority of transportation, and the number of transportation means (ambulances, helicopters, etc.) [23]. The required transport capacity (X) can be calculated with Equation (2.2) [23].

$$X = \frac{N \cdot t}{T \cdot n} \qquad (2.2)$$

X – Number of required transportation vehicles

N – Number of injured people

t – Distance of the hospitals to the place of action

n – Number of patients that can be transported at the same time

T – Entire travel time

Finally, Mohammed Khouj in the same research group has programmed an intelligent Reinforcement Learning agent to make resource allocation decisions. Khouj named his agent DAARTS (Decision Assistant Agent in Real Time Simulation) [11], [24]. Initially, DAARTS was programmed with Matlab. Then due to the large memory requirements of Matlab, the program was migrated to Java. At the time of writing this thesis, the Java version is still in development. The comparison in this thesis is performed with the Matlab version from Khouj's PhD proposal.

The benefits of using AI for modeling the human decision making process can be summarized as follows [11], [24]:

- More time efficient and improve the overall effectiveness
- Allows the modeling scenario to be run many times

- Can enable the user to custom make modeling scenarios and evaluate them
- The user does not need to interfere with the decision making process

Reinforcement has its roots in psychology. According to Myers, reinforcement is defined as "any event that strengthens, or increases the frequency of, a preceding response" [25]. A positive reinforcement may be a tangible reward, such as money or praise. In the context this thesis, the reward becomes the total patient discharge rate.

Reinforcement Learning is a type of learning algorithms in which an agent acquires knowledge or experience through interactions with its environment. The basic concept of Khouj's Reinforcement Learning can be summarized in three elements: state (input), action and reward [24]. The structure of the learning agent is shown in Figure 5. The agent first senses the state of the environment. Then the agent searches for that state in the Look Up Table (LUT). Figure 6 shows an example of an LUT. Each state in the LUT has a number of actions associated with it. The numerical values associated with a state/action pair are called Q-values. The number of Q-values in a LUT defines the size of the LUT. The agent is programmed to be greedy, meaning he will select the highest Q-value given a state. Once the agent selects a Q-value, he performs an action, which may be positive (a credit) or negative (a punishment). The purpose of RL is to learn a policy that dictates which actions, taken in which states, yield the greatest long-term reward. From the feedback of the reward value, the agent learns and re-iterates the process again. The Q-value receives an update at every iteration, as shown in Equation 2.3. The new Q-value is determined by adding the old Q-value to an incremental amount, as described in the equation.

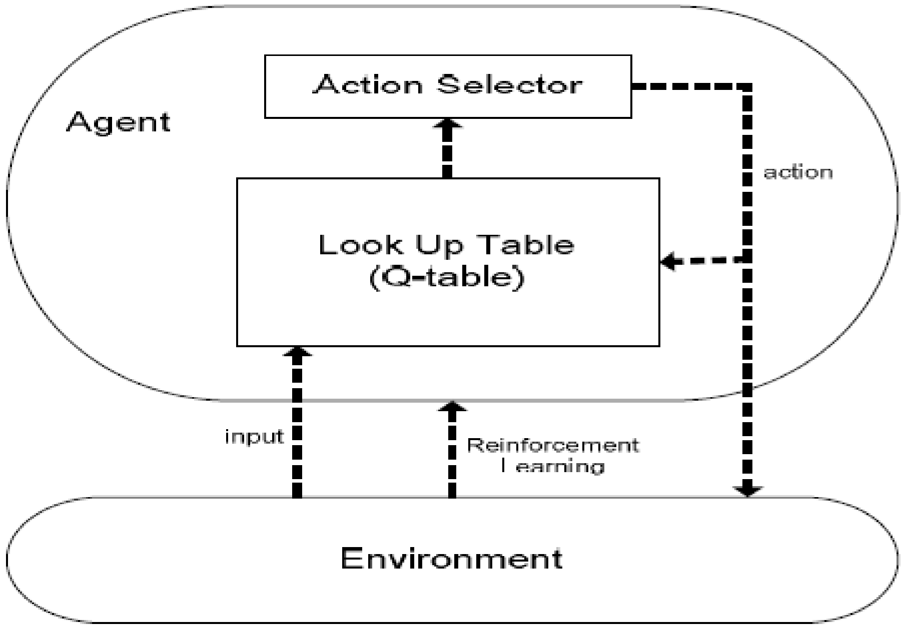

Figure 5: Structure of the Learning Agent [11]

(State, Action)	Q Value
(State 1, Action 1)	Q1
(State 1, Action 2)	Q2
…	…
(State n, Action n)	Qn

Figure 6: Sample Look Up Table

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \max(\gamma Q(s_{t+1}, a_{t+1})) - Q(s_t, a_t)] \qquad (2.3)$$

r_{t+1} – Agent's reward. In this thesis it means the total patient discharge rate.

α – Learning Rate. This parameter determines how fast learning takes place. A value of 0 means no learning takes place. A value of 1 means full learning takes place.

γ – Discount Factor. This parameter determines how much influence future rewards have on the learning process. A value of 0 for the discount factor means "opportunistic" decision making, without consideration for future value. A value of 1 means future values directly influence current value. Table 1 was prepared from the readings to show the extreme combinations of α and γ.

Table 1: Extreme Combinations of Learning Factor (α) and Discount Factor (γ)

	$\alpha = 0$	$\alpha = 1$
$\gamma = 0$	Agent does not learn.	Agent only thinks about the present state (opportunistic).
$\gamma = 1$	Agent does not learn.	Agent thinks about the long-term accumulation of awards.

In addition to the parameters defined in Equation 2.3, there is another parameter ε, the exploration rate. This parameter means every set number of steps, the agent takes a random action that would otherwise not be explored. Therefore this parameter helps the agent from falling into a suboptimal path. In the beginning, it is recommended to keep ε to a small number, in order to enable the agent to do more random exploring.

DAARTS follows the steps below to make decisions [11].

- The agent receives inputs from the environment and determines the state of the environment.
- Then the agent searches for that state in the LUT. There will be multiple (state, action) entries associated with that state.
- The agent chooses the (state, action) entry that corresponds to the largest $Q(s_t,a_t)$. Or picks it randomly if performing exploratory learning. The largest represents the action that is likely to lead to the most optimal number of discharged patients.
- The agent takes the corresponding action and determines the reward.
- Then the agent updates the Q-value associated with this (state, action) entry according to Equation (2.3). Over time, the LUT will converge to a set of Q-values that approach the actual optimal patient discharge rates that can be expected due to taking a specific action in a given state.

2.4 The Kaiser Building

As previously discussed, the initial focus of the thesis was building energy modeling, specifically the Fred Kaiser Building at UBC. The objective of this project was to examine where energy usage could be reduced and to then come up with a model for energy reduction. The energy system simulator of the Kaiser Building could then be coupled to the energy system optimization with the UBC campus using the I2Sim software. It would have required extensive electrical consumption data on Kaiser's energy usage to carry out this work. Due to the lack of metered data, the project had to be changed. In order not to waste the initial research work, this section is devoted to explaining the design of the Kaiser Building and how its energy modeling could be performed.

The Kaiser Building at UBC is home to the Department of Electrical and Computer Engineering. Construction of the building started in 2003 and it was finished in September of 2005 [26], [27], [28]. The cost of the construction was $18 million. The building construction had to address some key issues. These issues included spatial flexibility, environmental sustainability and contextual integration. Simulation tests performed on the building model show that the Kaiser building would consume 45% less energy compared to a traditional building [26], [27]. The Kaiser Building was constructed partially on top of the old CEME (Civil and Mechanical Engineering) building, and has a floor area of 96,000 ft^2 (8,900 m^2) spanning over five floors. The building houses 700 occupants, consisting of graduate students, faculty and staff members. Preliminary studies have shown the building is estimated to save $21,500/year of energy costs. The building features in-floor slab radiant heating and cooling [26]. Figure 7 shows a photo of the Fred Kaiser Building.

Figure 7: The Fred Kaiser Building

Energy efficiency is designed into the building. For example it was designed with dual flush toilets to improve the water efficiency. All the water urinals are waterless types. There are infrared activated water faucets. Overall that should reduce water consumption by over 50% compared to traditional fixtures. In terms of the construction material, 95% of the steel used were recycled. There is a high volume of fly ash concrete used, which reduced the production of Portland cement [28]. The windows of the building are also high performance, and designed to reduce the amount of ultraviolet rays entering the building.

Renewable sources are also designed into the building. The roof of the Kaiser Building has photovoltaic panels built in [29]. The generation capacity is 7 kW. Carmanah designed the photovoltaic system in partnership with Stantec Consulting. The system runs in parallel to UBC's local utility grid and is meant to supply the building with a significant fraction of its annual power load, estimated at 6.5 MW annually [29]. The 48 solar modules were sealed into double glazed windows and placed at the roof of the building.

In terms of the Heating Ventilation and Air Conditioning (HVAC) system, 75% of the Kaiser Building uses radiant heating and cooling [28]. The building is the 5th one in North America to be completed and operating using a large scale radiant heating/cooling slab system. There is a night-time operated cooling tower. The building has an induced demand-control natural ventilation system. There is also a CO_2 sensor controlled demand ventilation system.

According to Mr. Geoff McDonell, the mechanical designer who was with Omicron at the time, after the building was completed, there were some unforeseen problems. A Starbucks franchise was added to the atrium of the building. To save costs, instead of adding an entrance that would face the outside, the Starbucks shares the same entrance as the building. Starbucks is a popular coffee chain, and many people who walk by the building would stop by for a cup of coffee. Therefore, the double doors by the Starbucks were opened and closed much more than anticipated, causing a shift in building air pressure. By my count, in between classes, the doors probably open and close ten times a minute. That affects the building air pressures. As there is a frequent intake of cold air, the radiant slab system needs to work harder to keep the building at a constant temperature. This results in a significant increase in energy consumption.

Geoff McDonell also provided some commissioning data for the Kaiser Building. The energy costs saved per year is about $21,500. The greenhouse gases saved per year is about 2,500 tonnes. The total energy intensity is about 504 MJ/m^2/year [27].

The ICT building at the University of Calgary has a similar concrete slab structure and it was simulated using EnergyPlus [30]. The simulated and measured indoor temperature trends showed that EnergyPlus represented the ICT Building thermal and energy performance with reasonable accuracy.

To perform building energy modeling, one needs to have a checklist of basic things to look for. I have compiled a list, by no means exhaustive. The list includes:

- Type of building (institutional, commercial, residential, etc.)
- Number of occupants
- The main activity hours (i.e. 9 am – 5 pm)
- What are the main loads?
- What are the types of insulation available?
- How is the building heated?

Some potential modeling scenarios includes:

- Summer vs. winter
- Electrical Consumption variation throughout the day
- Heat retention
- Sun shines on different parts of the building

Parameters of the models could include:

- Surface area of the building (m^2)
- Floor area of building (m^2)
- Volume of the building (m^3)
- Thermal conductivity of the material (W/(m*K))
- Cost of electricity ($/kWh)
- Number of occupants
- Number of sunlight hours
- Time of day (morning, afternoon, evening, etc.)
- Starting temperature of building (initial condition)

I initially looked into applications of optimization in building energy usage. However, after some time, it was found that the Kaiser Building electrical consumption is in fact not separately metered. Without any real usage data, it became difficult to proceed with this project in a meaningful way.

2.5 Building Energy Simulation Software

I also did some research into the main building energy simulation software available on the market, as mentioned by [30], to determine which might be a good fit for modeling the Kaiser Building. This section discusses advantages and disadvantages of the software.

The first energy simulation software I looked into was ESP-r. According to its website, ESP-r is an open-source integrated energy modelling tool for the "simulation of the thermal, visual and acoustic performance of buildings and the energy use" [31]. It also has associated environmental control systems. The system is equipped to model heat, air, moisture and electrical power flows at user determined resolution. ESP-r is designed for the Unix operating system, but it can be used with Windows [31]. In terms of cost, I did not find any cost information for ESP-r.

TAS is a software package developed by US Department of Energy. It is used for the thermal analysis of buildings. TAS includes a 3D modeller, a thermal/energy analysis module, a systems/controls simulator and a 2D Computational Fluid Dynamics (CFD) package. Strengths of TAS include excellent response and its accuracy for concept development. Weaknesses of TAS include the fact that it is not intended for detailed services layout design. There is a free trial CD-ROM available. The cost for a license is ₤1600 ($2,500 Canadian) for a 1-year license [32].

According to its official website [33], TRNSYS (Transient System Simulation Tool) is an "extremely flexible graphically-based software environment used to simulate the behavior of transient systems". TRNSYS is composed of two main components. The first component is an engine (kernel) that reads and processes the data in the input file. The second component is an extensive library of parts; each models the performance of one part of the system. In terms of pricing, customers can purchase either the full software package or individual libraries. The education price for the full package is $2370. According to the ICT Building paper, there is a RC-conduction transfer model for TRNSYS. But it is not applicable to radiant panel systems and may not be accurate in simulating radiant slab systems [33].

IDA ICE 4 is a dynamic multi-zone simulation application for accurate study of thermal indoor climate of individual zones [34]. It is also capable of modeling the energy consumption of an entire building. There is no publicly available pricing information for IDA ICE on its website

[34]. Also according to the ICT Building paper, IDA ICE has a very small user base and there is limited literature published [30].

EnergyPlus is a whole building energy simulation program that engineers, architects, and researchers use to model energy and water use in buildings [30], [35]. In terms of cost, there is no cost except for a commercial source license. EnergyPlus is also the software chosen by a PhD student at the University of Calgary to simulate the ICT Building [30]. Simulation results agreed well with the field measured data for both cooling energy use and indoor operative temperatures. But simulation results were sensitive to construction and system parameters [35].

With the overall criteria of cost, functionality, and learning curve, my opinion is EnergyPlus may be best suited to model a radiant concrete slab building such as the Kaiser Building.

2.6 Building Energy Policies

In addition to the Kaiser Building, I also did some research into energy policies pertaining to buildings. Any high level decision making would not be possible without knowing the policies. Energy policy in particular is very important as it guides how energy management systems are designed and implemented. A properly designed energy policy would help yield maximum benefits. Therefore it would be crucial to examine some of the relevant policies.

2.6.1 UBC's GHG Reduction Goals

The University of BC has set aggressive goals to reduce its greenhouse gas emissions. The university has named the project "Living Lab", using the university campus to experiment with sustainability. The project includes replacement of the existing steam heating plant with hot water heating. The broad targets of Living Lab are as follows [6], [7]:

- By 2015, the university is to reduce the Greenhouse Gas (GHG) emissions by 33% from 2007 levels.
- By 2020, the university is to reduce the GHG emissions by 67% from 2007 levels.
- By 2050, the university is to reduce GHG emissions by 100% from 2007 levels by 2050 (net positive energy producer).

Recently, the university began another phase of the Living Lab project by identifying a list of buildings to monitor energy consumption.

2.6.2 BC Energy Plan

The BC government has set ambitious goals to reduce its energy demands and become self-sufficient in the production of electricity. The BC Energy Plan sets targets that are to be reached. According to the plan, it requires 50% of BC Hydro’s incremental growth demand through energy conservation by 2020. That means 10,000 GWh (Gigawatt Hour) of the forecast load will be met through demand reduction. BC Hydro will be aggressively pursuing and exceeding its existing target [36].

For new buildings, the BC Energy Plan states:

“New provincial public sector buildings will be required to integrate environmental design to achieve the highest standards for greenhouse gas emission reductions, water conservation and other building performance results such as a certified standard.” [36]

Therefore, that means buildings are a major consumer of energy and there is much potential for energy savings to be made in the buildings. In commercial and institutional buildings with high people traffic, a set of double doors are incorporated to help stabilize the air pressure inside the building. Also in the Kaiser Building, the building occupancy was overestimated by including the undergraduate students. Overall, the Kaiser Building project was still a positive learning experience and I gained a better understanding of complex interdependent systems.

3 Proposed Lagrange Based Optimization Algorithm

This chapter discusses the proposed Lagrange Based Optimization (LBO) algorithm. For our problem of allocating resources to hospitals, there are a number of challenges. The main challenges include: a huge set of possible solution space, and functions with discrete levels. After careful consideration, an approach based on the Lagrange Multiplier was adopted. In terms of scalability, this algorithm is very scalable. As number of inputs increase, it can still provide accurate results.

3.1 What is Optimization?

Optimization is defined as the process of finding the maximum or minimum value of a function or problem, with or without constraints. Everyday examples of optimization include how to drive from point A to point B in the shortest time; how to maximize the amount of work performed and how to minimize wait time at a popular restaurant. Other examples of optimization include cost minimization, profit maximization and patient output maximization for a hospital. In this thesis, the main objective function is to maximize the number of patients discharged from the hospitals.

Broadly speaking, there are several types of optimization problems, including [37]:

- Unconstrained Optimization
- Equality-constrained Optimization
- Inequality-constrained Optimization

The type of optimization in this thesis is equality-constrained optimization. The decision maker is given certain amounts of water and electricity to distribute to the hospitals.

3.2 The Lagrange Multiplier

The Lagrange Multiplier is a common and useful mathematical method to solve constrained optimization. It was named after French mathematician Joseph Louis Lagrange. Simply speaking, in Lagrange's method, the problem is optimizing an objective function given constraints. A new variable called λ or the Lagrange multiplier is introduced. The purpose of λ is to help solve the first order conditions.

According to Dixit and Baldick, Lagrange's Theorem in a continuous domain is defined as follows [38], [39]. Suppose x is a two-dimensional vector defined as $[x_1\ x_2]$, c is a scalar, and F(x) and G(x) scalar functions. F(x) is the objective function and G(x) is the constraint. Define the function L as in equation 3.2. If x* maximizes F(x) subject to G(x) = c, with no other constraints (such as non-negativity), and if $G_j(x^*) \neq 0$ for at least one j, then there is a value of λ such that:

$$L_j(x^*,\lambda) = 0 \text{ for } j = 1,2 \ L_\lambda(x^*,\lambda) = 0 \qquad (3.1)$$

$$L(x,\lambda) = F(x) + \lambda[c - G(x)] \qquad (3.2)$$

Then take the partial derivatives or the First Order Conditions (FOC) of the objective function, with respect to each of the variables, and set them to zero. The FOCs are:

$$\frac{\partial L}{\partial x_1} = 0 \quad (3.3)$$

$$\frac{\partial L}{\partial x_2} = 0 \quad (3.4)$$

$$\frac{\partial L}{\partial \lambda} = 0 \quad (3.5)$$

The interpretation of the Lagrange Multiplier λ is the sensitivity of the Lagrange function to changes in the constraint [40]. It can also be interpreted as the sensitivity of the optimal value of the objective function to variation in the constraints [41].

In his paper, Everett mentions Lagrange multipliers are "well suited to the solution of problems of allocating limited resources among a set of independent activities" [42]. The application of the Lagrange multipliers does not guarantee a result will be found for every case, but it is "fail-safe" because any solution found will be the optimum. Compared to other methods, the Lagrange multiplier method is very simple, and it is worth trying first.

There is also a mathematical proof that the solution found is optimum. The notation used in Everett's paper is adapted to be consistent with the notation used in this thesis [42]. Suppose if there is a vector x* in the solution space S, that maximizes:

$$F(x) + \lambda[c - G(x)] \quad (3.6)$$

Then that means, for all xεS,

$$F(x^*) + \lambda[c - G(x^*)] \geq F(x) + \lambda[c - G(x)] \ (3.7)$$

Rearranging this, we have:

$$F(x^*) \geq F(x) + \lambda[G(x^*) - G(x)] \ \ (3.8)$$

For all xεS.

If this latter inequality is true for all xεS, then it also holds true for any subset of S. On the subset S*, the term $\lambda[G(x^*) - G(x)]$ is non-negative by definition of the subset and the non-negativity assumption of λ. Therefore, the inequality reduces to $F(x^*) \geq F(x)$, and the theorem is proved [42].

In the context of this thesis, the objective function is to maximize the number of discharged patients. The constraints are limited resources of water and electricity. The hospitals HRTs are assumed to have a concave down curve when plotted. In addition to water and electricity, there are also other input constraints that limit the capacity of the hospitals. These inputs can be health care personnel (doctors, nurses, etc.), medicines, medical supplies and physical space of the hospitals. These factors taken into consideration mean that the HRTs cannot increase uncontrollably. This concave down requirement is to ensure a maximum is found. If the curves were concave up, then any bounds on the function would only produce a minimum. The hospitals' capacity to discharge patients cannot increase infinitely with more resources added. At some point it will saturate. Due to the nature of HRTs, we are dealing with a discontinuous step function. The objective function is defined as follows in Equation 3.9, where n is the number of hospitals.

$$\text{Max} \sum_{i=1}^{n} N_i = N_t \qquad (3.9)$$

N_i – number of patients discharged by hospital i

N_t – total number of patients discharged

Note N_i is a function of P_i and W_i, the power and water delivered to each hospital.

The constraints are shown in Equations 3.10 and 3.11:

$$\sum_{i=1}^{n} P_i = P_t \qquad (3.10)$$

$$\sum_{i=1}^{n} W_i = W_t \qquad (3.11)$$

P_i – power supplied to a hospital

P_t – total power available

W_i – water supplied to a hospital

W_t – total water available

Also each P_i and W_i must be non-negative.

After formulation of the problem we augment the constraints in objective function by using the Lagrange multipliers. A new function is set up by adding two Lagrange multipliers to the original objective function. The augmented function is shown as follows in Equation 3.12. Note this is only possible for continuous case.

$$\mathcal{L} = N_t - \lambda_1 \left(P_t - \sum_{i=1}^{n} P_i \right) - \lambda_2 \left(W_t - \sum_{i=1}^{n} W_i \right) \qquad (3.12)$$

Then we take the partial derivatives of the augmented function and set each one equal to zero. That provides the necessary first order conditions to solve the problem. The maximum of this function is found at the point where the partial derivatives of the function to its variables are equal to zero.

Equations 3.13 and 3.14 show the First Order Conditions of Equation 3.12.

$$\frac{\partial \mathcal{L}}{\partial P_1} = \frac{\partial N_1}{\partial P_1} - \lambda_1 = 0, \dots, \frac{\partial \mathcal{L}}{\partial P_n} = \frac{\partial N_n}{\partial P_n} - \lambda_n = 0 \qquad (3.13)$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = \frac{\partial N_1}{\partial W_1} - \lambda_2 = 0, \dots, \frac{\partial \mathcal{L}}{\partial W_n} = \frac{\partial N_n}{\partial W_n} - \lambda_n = 0 \quad (3.14)$$

After solving for the different distributions of water and electricity, the answers can be substituted into the objective function to find the maximum number of treated patients. At last we can integrate the solution back into the system and check for the resource interdependencies.

The limitations of the Lagrange Based Optimization method include:

- There is a concavity requirement for the objective function. A concave down function with the resource constraints forming an upper bound will ensure the maximum is found. Otherwise the resource constraints would form a lower bound for the objective function and only the minimum is found.
- It does not guarantee an answer can be found in every case, but if one is found it will be the optimum [42].

3.3 Applications in Power Systems

Lagrange multipliers can also be used to solve power system problems. One application area is the Automatic Generator Control/Economic Dispatch Control (AGC/EDC) problem. In an energy management system, the AGC program controls the electrical power output of generators so as to supply the continuously changing customer power demand in an economical manner [43]. The power system dispatcher also plays an important role, because he interacts with the program to incorporate the current operating conditions. According to Momoh, the basic objectives of power system operation during normal operating conditions associated with AGC are the following [43].

- Ensure the total amount of power generated matches the total amount of power consumed at the loads
- Minimize the power system's electrical frequency error, preferably to zero
- Allocate the amount of power generated among the control areas to ensure the actual net area tie power flows to match the scheduled amounts

- Minimize the area operating costs by distributing the area generation among the generation sources

The first objective is usually associated with governor speed control. The second to fourth objectives are accomplished by supplementary controls directed from area control centers [43].

Economic Dispatch Control (EDC) allocates generation outputs of the committed generating units in order to minimize the fuel costs, while meeting system constraints such as spinning reserve. The EDC functions to compute recommended economic base points for all manually controlled units as well as economic base points for units which may be controlled directly by the EMS (Energy Management System) [44].

As an example, please consider the following EDC problem [45]. The objective function is to minimize the total generation cost from the three power plants, as shown in Equation 3.15. Equation 3.16 provides the equality constraint that the total generation must be 800 MW. Equations 3.17 to 3.19 define the costs functions for the three power plants. P_1, P_2 and P_3 denote the power generated by the three plants, respectively. The costs are in $/MWh.

$$Min\ C_1 + C_2 + C_3 \qquad (3.15)$$
$$subject\ to\ P_1 + P_2 + P_3 = 800\ MW \qquad (3.16)$$
$$C_1 = 500 + 5.3P_1 + 0.004P_1^2 \qquad (3.17)$$
$$C_2 = 400 + 5.5P_2 + 0.006P_2^2 \qquad (3.18)$$
$$C_3 = 200 + 5.8P_3 + 0.009P_3^2 \qquad (3.19)$$

The augmented function is shown in Equation 3.20.

$$L = C_1 + C_2 + C_3 + \lambda(800 - P_1 - P_2 - P_3) \qquad (3.20)$$

The First Order Conditions are shown in Equations 3.21 to 3.24. They are taken with respect to the independent variables P_1, P_2, P_3 and λ.

$$\frac{\partial L}{\partial P_1} = 5.3 + 0.008P_1 - \lambda = 0 \qquad (3.21)$$
$$\frac{\partial L}{\partial P_2} = 5.5 + 0.012P_2 - \lambda = 0 \qquad (3.22)$$
$$\frac{\partial L}{\partial P_3} = 5.8 + 0.018P_3 - \lambda = 0 \qquad (3.23)$$
$$\frac{\partial L}{\partial \lambda} = 800 - P_1 - P_2 - P_3 = 0 \qquad (3.24)$$

After solving the above system of four equations, the results are obtained.

P_1 = 400 MW
P_2 = 250 MW
P_3 = 150 MW
$\lambda = \$8.5\ /MWh$

This provides a total cost of \$3260 + \$2150 + \$1272 = \$6682/hour. Figure 8 shows the plot of the three plants and the unit generation cost. One critical point to note is all three power plants produce the optimal amounts of power at their intersection with the horizontal line λ. If they are not producing at the same λ, the extra cost incurred in one plant would be greater than the cost savings incurred in another. That would not produce the minimum total cost.

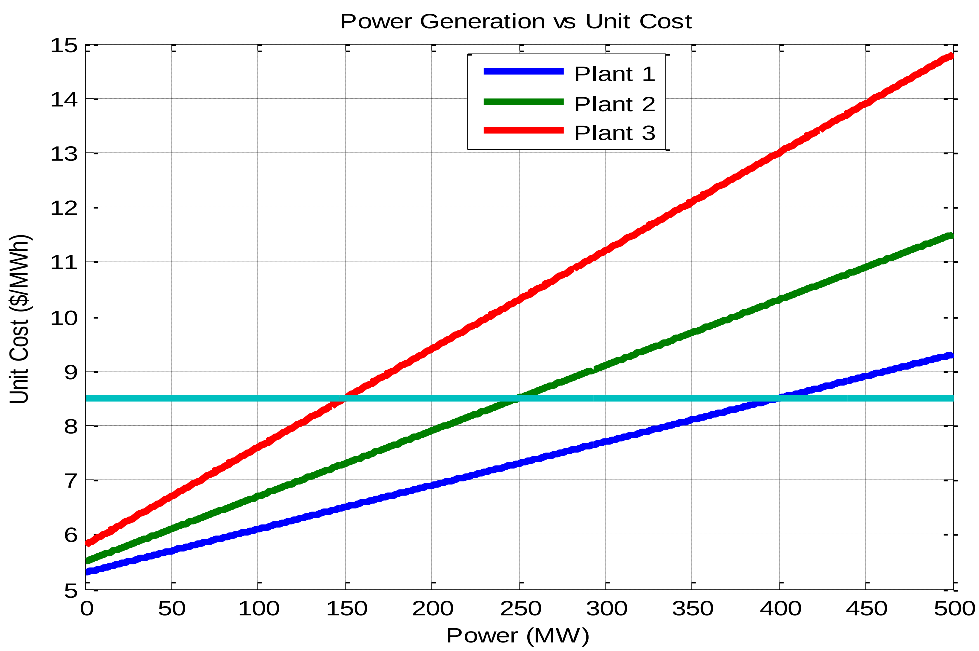

Figure 8: Power Generation vs. Unit Cost

3.4 How to Build the HRTs

One of the key questions is how to assemble the HRTs? As mentioned before, I2Sim is a simulation tool and it relies on the user to input HRTs. A realistic and accurate HRT will help produce reliable results. One of the best ways to obtain the input data is to talk to the experts who operate the production units in real life. The data from the HRTs can be obtained from interviews with hospitals managers, utility operators, emergency responders and other infrastructure operators. Based on their experience, they can tell the researcher the number of resources and their quantities required for a hospital to be fully operational. For example, the best hospitals in the Vancouver area are capable of discharging ten patients per hour. If a hospital can build an appropriate HRT for different situations, and pass them to EOC, then good decisions can be made. In this thesis, the hospitals' HRTs from a modeling scenario are modified data obtained from a confidential Olympics report, in order to protect the sensitive information.

3.5 Summary of the Lagrange Based Optimization Method

The LBO Method used to calculate the optimal dispatches for scenarios in Chapter 4 can be summarized as follows.

- Find out the HRTs of all the hospitals
- Find out the available amounts of resources (e.g. electricity and water). The assumption is other necessary resources such as health care personnel, medications, medical supplies are always sufficient.
- Calculate the operational efficiencies of the hospitals based on electricity, and find the rows the hospitals should operate on.

- Calculate the operational efficiencies of the hospitals based on water, and find the rows the hospitals should operate on.
- Compare solutions of electricity and water, and resolve any conflicts (water limiting, electricity limiting, etc.). Then suggest best solution.
- Use the optimal solution to calculate the number of people treated
- *This can be expanded for more resources

Figure 9 shows a flowchart of the LBO method when programmed into Matlab. The LBO method is assumed to have a two dimensional solution in this case: water and electricity. The number of dimensions would increase with more resources added. In general, there would also need to be a network of sensors to detect the state of the system.

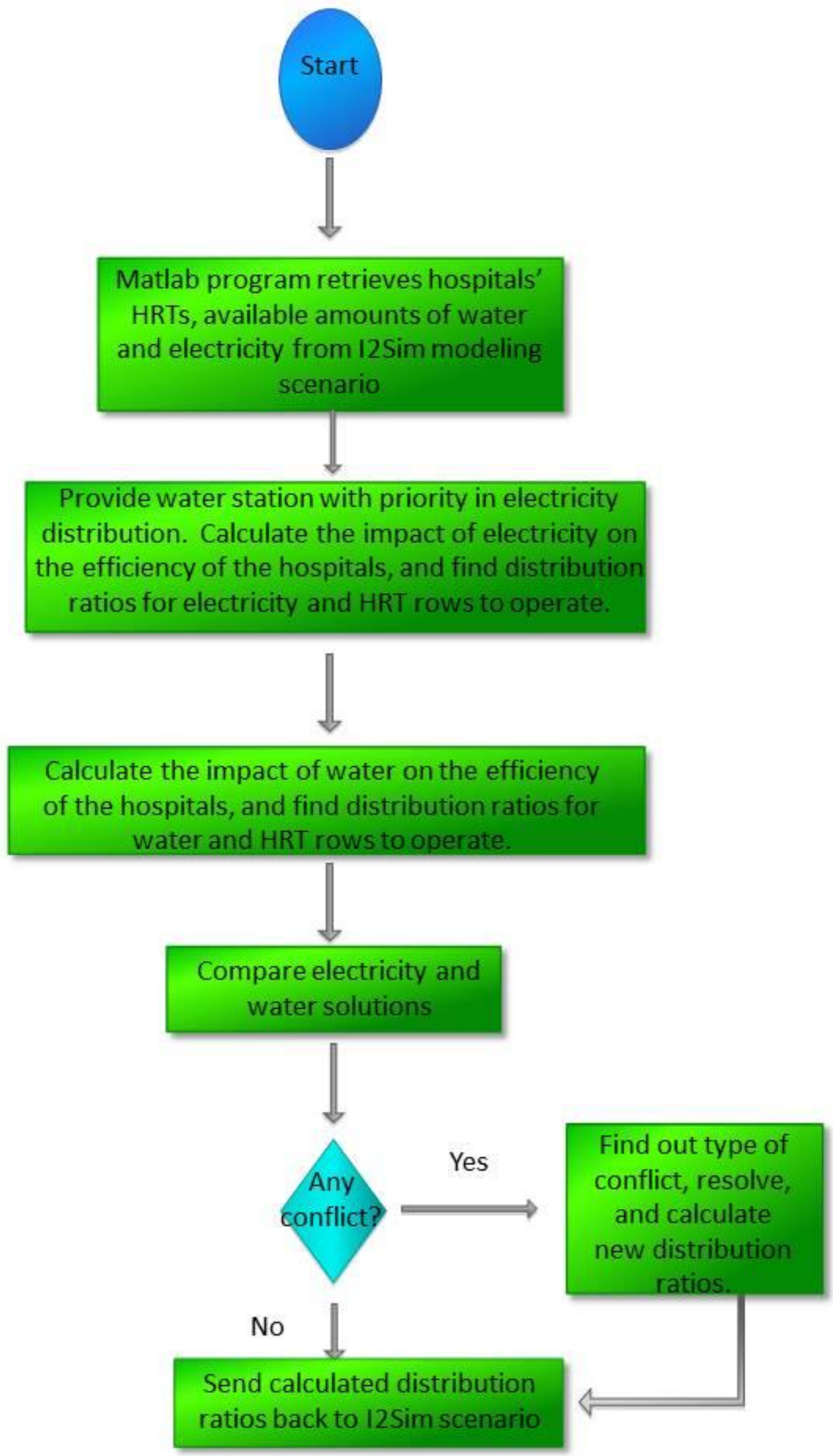

Figure 9: Lagrange Based Optimization Method Flowchart

4 Simulation Scenarios and Results

This chapter discusses the I2Sim simulation scenarios and their results. In the first section there is a scenario of three hospitals with no transportation. The objective of this scenario is to discharge the maximum number of patients from the hospitals. The second section discusses a scenario which consists of three venues and three hospitals. The objective functions are to transport the maximum number of patients to the hospitals, and to discharge the maximum number of patients from the hospitals. The third section tests the three hospitals with no transportation again, but this time with the RL agent method. The final section of compares the results obtained by the two methods, and examines the advantages and disadvantages of each. In all simulations, there are one PM and five RMs.

4.1 Three Hospitals Case No Transportation with LBO Method

To demonstrate the Lagrange Based Optimization algorithm, it is first shown for a simple three hospitals scenario with no transportation. By no transportation, it means that each hospital already has a waiting room full of patients. Figure 10 shows a conceptual diagram for the resource interdependencies for a hospital [46]. Each hospital's optimal performance depends on receiving adequate quantities of both water and electricity. It is assumed that other inputs needed are always sufficient. The modeling scenario I built with I2Sim is shown in Figure 11. The resource distributions are calculated only once at the beginning of the simulation, and it is assumed no event occurs during the 24-hour simulation period. On the left of the figure, the two cells represent an electrical substation and a water pump station. The I2Sim Control Panel enables the user to specify the length of simulation, time units and step sizes. In this section and the third section of this chapter, the simulation time is 24 hours. The I2Sim Visualization Panel enables the user to specify which probe(s) to display. The water pump station requires some electricity (or power) to operate, giving a rise to interdependency. The electricity sent to the water pump station would be used to drive the motors and other electrical equipment inside. The sources (BC Hydro and Water Supplier) are assumed to be able to supply a constant amount of water and electricity. The three production cells in the middle of the figure represent the three hospitals. The hospitals are each attached to a waiting room full of patients (10,000 patients), and the hospital cell provides a command signal of how fast the patients are discharged. After that the patient outputs are summed by the aggregator.

As mentioned before, the overall objective is to maximize the number of patients discharged by the three hospitals. There may be limited resources of water and electricity and the distributors need to decide how to optimally distribute the resources.

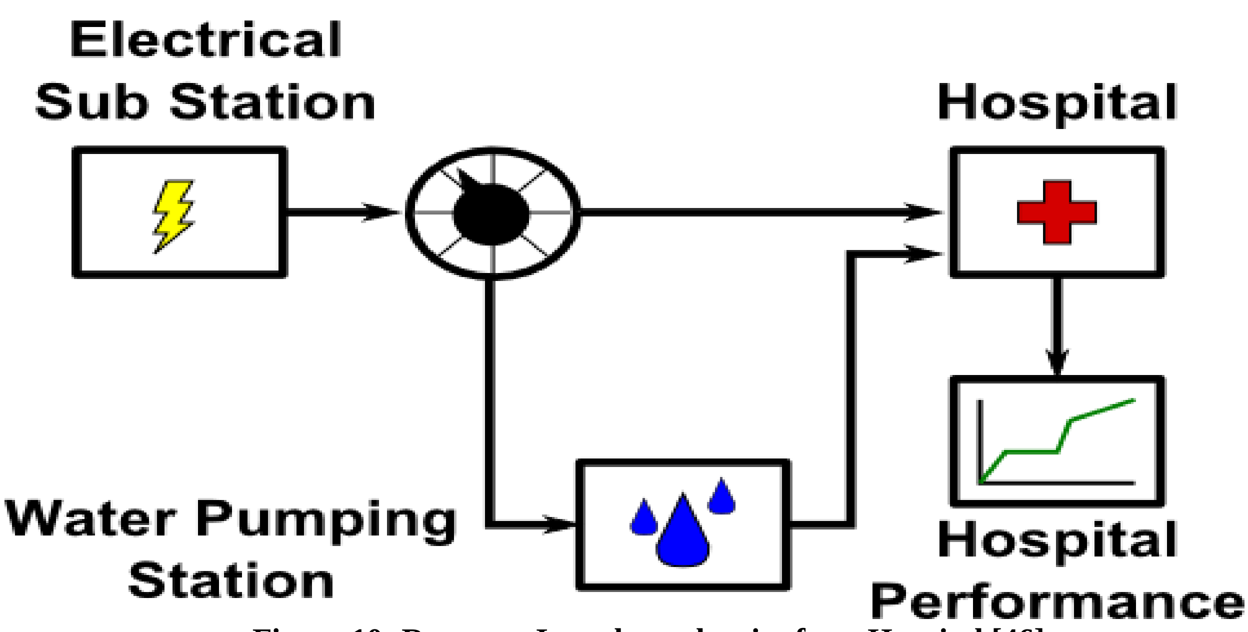

Figure 10: Resource Interdependencies for a Hospital [46]

One thing to note is that electrical substations are designed to distribute discrete combinations of power. That is, they are not like a knob on an oven where the decision maker can decide the fine difference between giving 20.5% or 21%. Depending on the number of feeders available, one can only switch on and off the available number of feeders to distribute power. If there is only one feeder available, that means a hospital may either get 100% or 0% of its required power. This is one limitation of this simulation that does not quite capture the real world.

On the demand side, one can use Non-Intrusive Load Monitoring (NILM) to monitor the amount of usage [47]. With Smart Meters, they may help control the load by shedding certain appliances. If the lines are being overloaded, the Smart Meter may for instance turn off the oven inside a house to help conserve power.

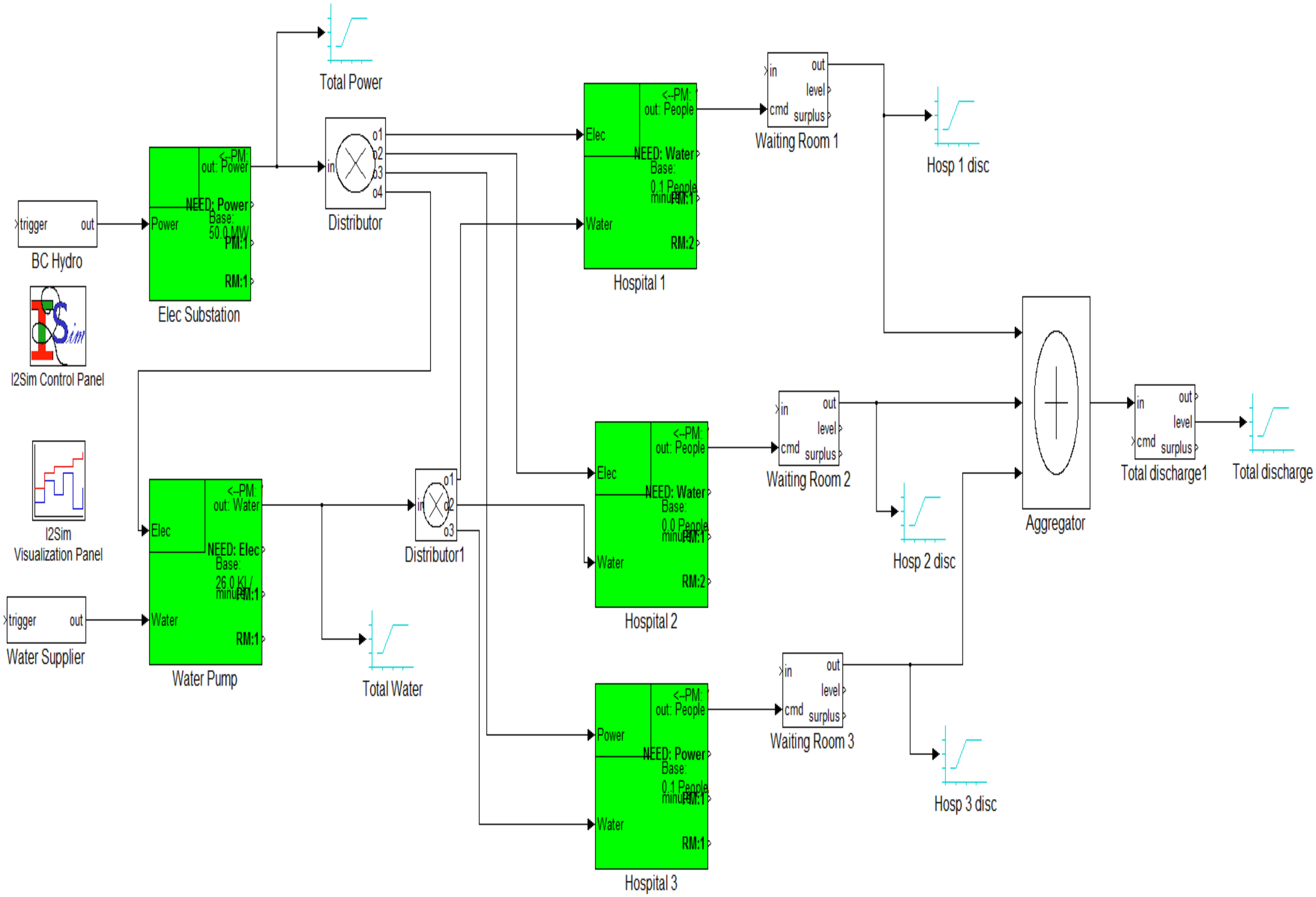

Figure 11: I2Sim Three Hospitals Scenario with No Transportation

Tables 2 to 6 show the HRTs for the three hospitals, the water pump station and the electrical substation. The patient outputs are shown in per hour time interval and rounded to the nearest integer. Since patients can only be discharged in integer numbers, the patient outputs in some rows have been rounded to the same number, such as rows 1 and 2 of Table 2. The actual patient outputs have been put in double asterisk after each table. The three hospitals presented in this scenario are less efficient hospitals compared to the best hospitals in the Vancouver area, which according to a confidential Olympics report are capable of discharging 10 patients per hour. The more efficient hospitals are used in the second scenario. The patient discharge rates are not necessarily linear over the one hour period. But the decision maker can extract the above into the simulation data.

The unit for power or electricity is kilowatts (kW). The unit for water is kiloliters/hour (kL/hour). The decisions to be optimized are how much water and electricity to distribute to each hospital. Since each hospital has different capabilities, distributing the resources equally to the three hospitals would not result in the optimal patient discharge.

Table 2: Hospital 1 HRT

Row Number	Patient Output (Patients/hour)**	Electricity Input (kW)	Water Input (kL/hour)
1	6	10000	360
2	6	7500	270
3	5	5000	180
4	3	2500	90
5	0	0	0

** Actual patient outputs (patients/hour) are: 6 (Row 1), 5.7 (Row 2), 5 (Row 3), 3 (Row 4), 0 (Row 5)

Table 3: Hospital 2 HRT

Row Number	Patients Output (Patients/hour)**	Electricity Input (kW)	Water Input (kL/hour)
1	1	5000	240
2	1	3750	180
3	1	2500	120
4	1	1250	60
5	0	0	0

** Actual patient outputs (patients/hour) are: 1.25 (Row 1), 1.17 (Row 2), 1 (Row 3), 0.58 (Row 4), 0 (Row 5)

Table 4: Hospital 3 HRT

Row Number	Patient Output (Patients/hour)**	Electricity Input (kW)	Water Input (kL/hour)
1	5	16000	960
2	5	12000	720
3	4	8000	480
4	2	4000	240
5	0	0	0

** Actual patient outputs (patients/hour) are: 4.8 (Row 1), 4.6 (Row 2), 4 (Row 3), 2.4 (Row 4), 0 (Row 5)

Table 5: Water Pump Station HRT

Row Number	Water Output (kL/hour)	Electricity Input (kW)	Water Input (kL/hour)
1	1560	10	1560
2	1200	8	1200
3	780	5	780
4	420	3	420
5	0	0	0

Table 6: Electrical Substation HRT

Row Number	Output Electricity (kW)	Input Electricity (kW)
1	32000	32000
2	24000	24000
3	16000	16000
4	8000	8000
5	0	0

In this scenario, it is also assumed there are no policies placing restrictions on how to distribute resources to the hospitals. For instance, a policy might state a particular hospital needs minimum amounts of water and electricity. This would change the decision.

Water is a required resource for the hospitals to operate. The amount of electricity required by the water pump station is small and it needs to be given priority in electricity distribution. Therefore the water pump station should always be given the first row of electricity (10 kW) in its HRT. After the water pump station has been given the required amount of electricity to operate, that amount is subtracted from the available amount of electricity.

Figures 12 and 13 illustrate the electricity and water input vs. the number of treated patients for the three hospitals, respectively. These plots are made from the inputs and output columns of the hospital HRTs. The input columns are electricity and water. The output column is the patient output. As can be seen from the figures, Hospital 1 is the most efficient hospital. Hospital 3 is the second most efficient hospital. Hospital 2 is the least efficient hospital. The LBO algorithm satisfies the resource needs of the most efficient hospital first. In I2Sim, since the thresholds need a minimum amount of resources to be triggered, this strategy would ensure the highest thresholds are triggered with the available amounts of resources.

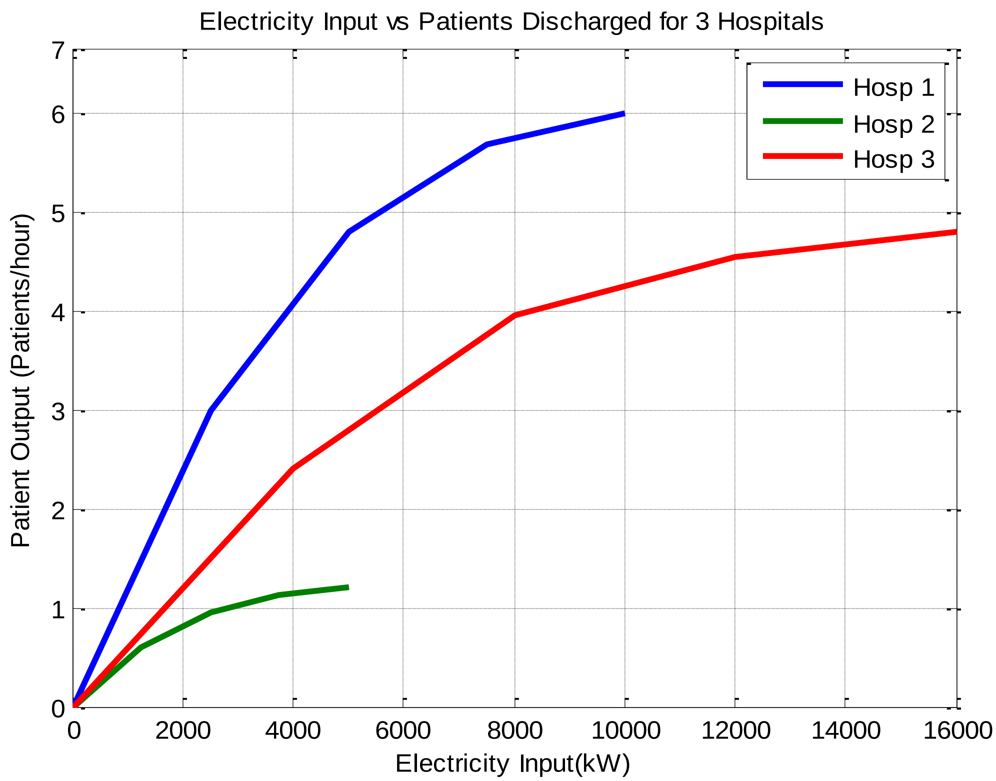

Figure 12: Electricity Input vs. Number of Treated Patients for the Three Hospitals

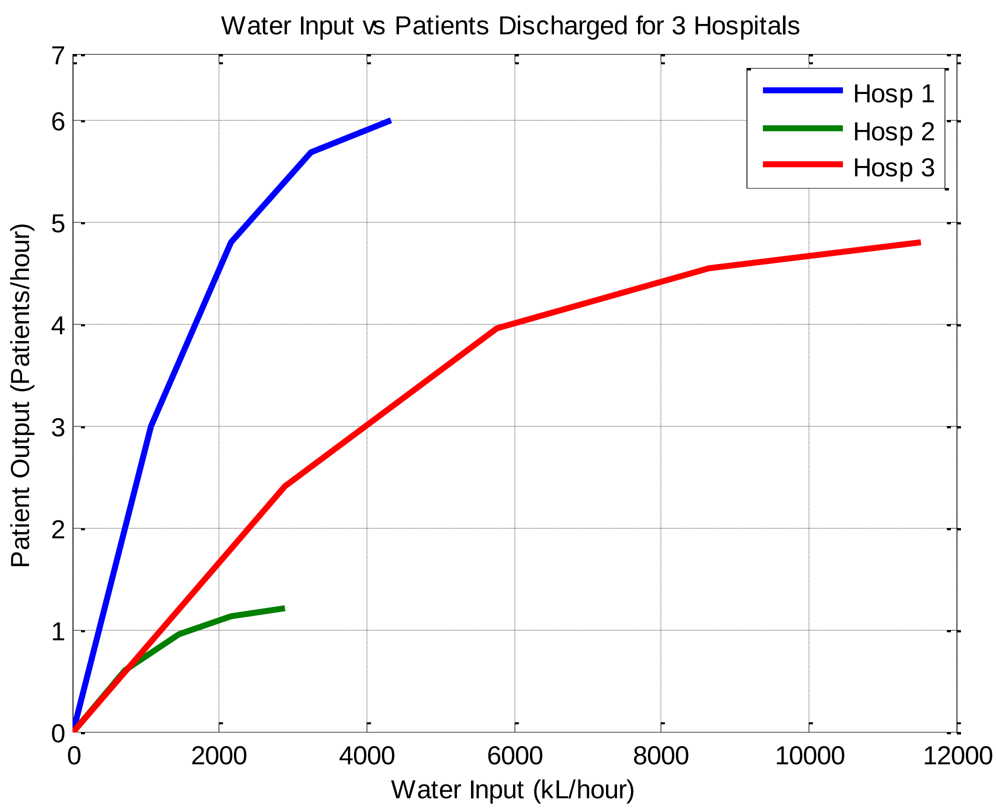

Figure 13: Water Input vs. Number of Treated Patients for the Three Hospitals

After the plots of electricity and water vs. patient discharged are produced, the next step is to find out the operational efficiencies (λ). These are the slopes of the line segments of the plots. They provide the (patients/hour)/kW and (patient/hour)/(kL/hour). These efficiencies are ranked and provide information on how efficiently the hospitals operate. Figure 14 show the operational efficiency versus availability of electricity. Figure 15 show the operational efficiency versus availability of water. Equations 4.1 and 4.2 show how the efficiencies are calculated.

$$\lambda_{water} = \frac{Patients/hour}{Water\ (kL/hour)} \qquad (4.1)$$

$$\lambda_{electricity} = \frac{Patients/hour}{Electricity\ (kW)} \qquad (4.2)$$

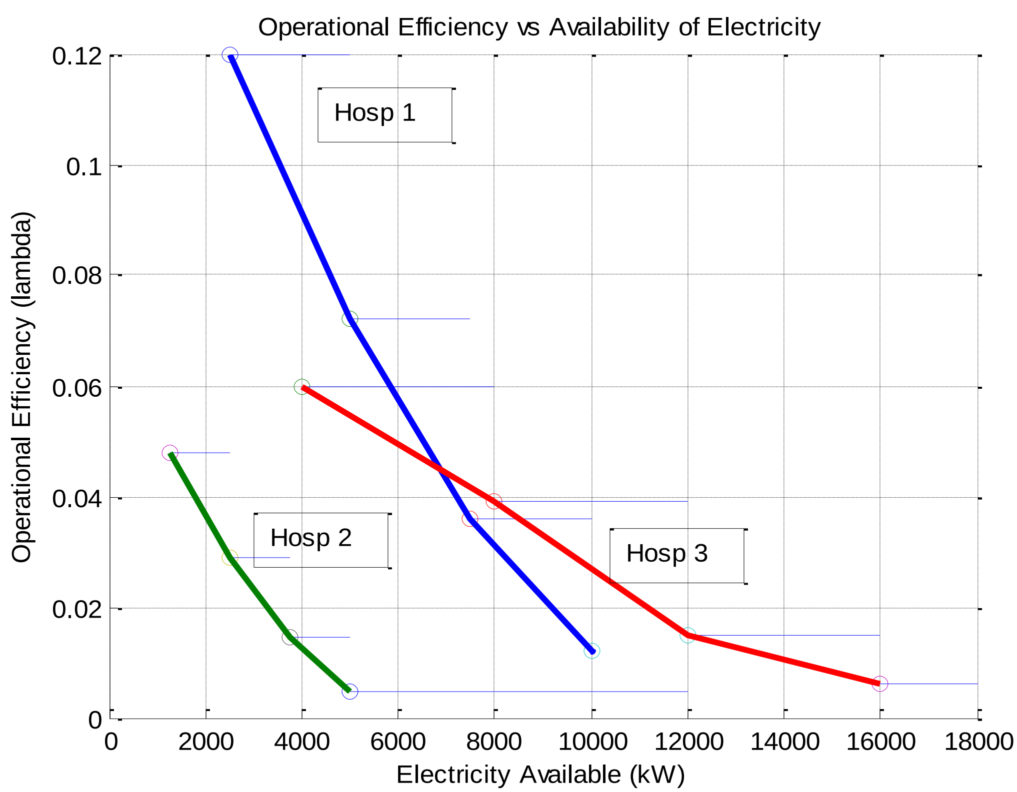

Figure 14: Hospital Operational Efficiency (λ) vs. Availability of Electricity

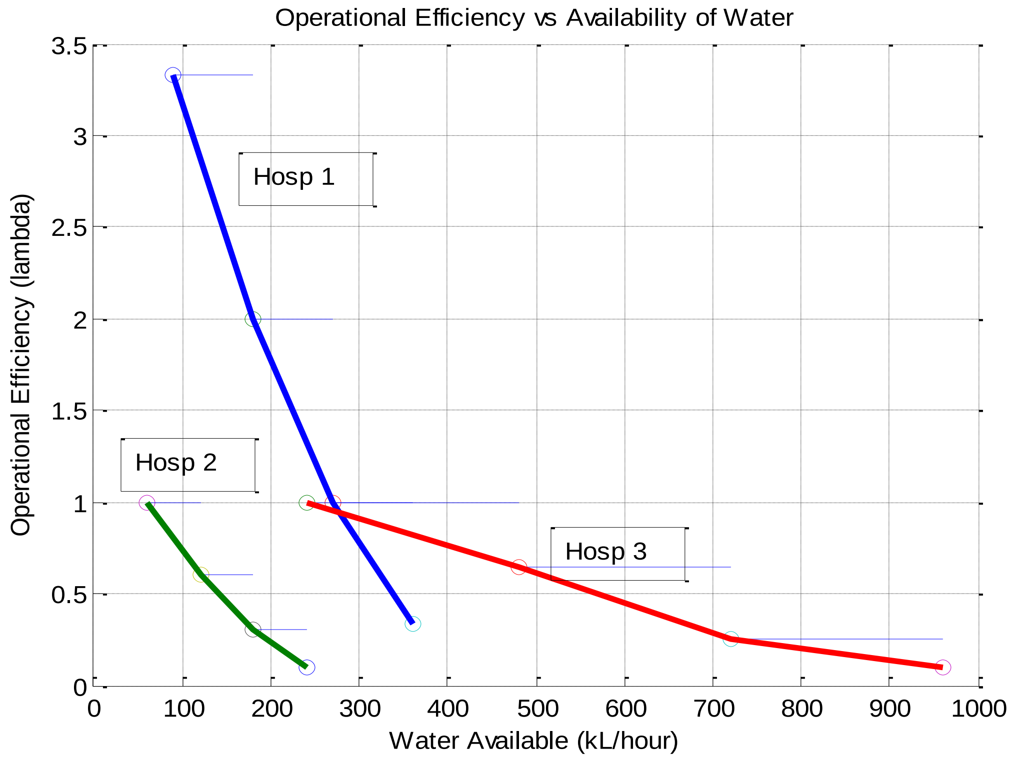

Figure 15: Hospital Operational Efficiency (λ) vs. Availability of Water

Table 7 shows the electricity distribution given various amounts of available electricity (after subtracting the amount required by the water station). Table 8 shows the water distribution. R in the tables denotes row. For example R4 means Row 4 of a hospital's HRT. The Matlab program used to build these tables can be found in Appendix A. The electricity assignment process has the following pseudo-code steps. The water assignment process is similar.

- Build a vector of all the electricity efficiencies and their associated hospital, ranked from top to bottom.
- Find out the amount of electricity available.
- Go down the list of efficiencies and assign values.

- If the corresponding electricity to this efficiency < available electricity + electricity already assigned to this hospital, AND if no electricity has been assigned to this hospital yet

    -then assign this amount of electricity to the hospital

- Else if an amount has already been assigned to this hospital

    -First give the assigned amount back to the available amount of electricity

    -Then assign this hospital the correct amount

- Update amount of available electricity left
- Iterate through this process until no electricity is left or the amount of electricity left is not sufficient to be assigned to any hospital

This process can be visualized as a horizontal line that starts scanning from the top of the operational efficiency plot and moves downward. There is a number attached to the line, the amount of electricity or water available. Whenever the line reaches a λ point, it finds out the corresponding electricity and the hospital to this point and makes an assignment. This horizontal line keeps moving down the plot until there is not enough electricity left to make another assignment. Please note due to the HRTs of the electrical and water stations, all outputs from the water and electrical stations are discretized to five levels only. Therefore, it is not possible to show all of the distributions on an I2Sim model.

Table 7: Electricity Distribution

Available Electricity (kW)	Hospital 1 Distribution (kW)	Hospital 2 Distribution (kW)	Hospital 3 Distribution (kW)	Total Patient Output (Patients/hour)
0 – 1,240	0 (R5)	0 (R5)	0 (R5)	0
1,250 – 2,490	0 (R5)	1,250 (R4)	0 (R5)	1
2,500 – 3,740	2,500 (R4)	0 (R5)	0 (R5)	3
3,750 – 4,990	2,500 (R4)	1,250 (R4)	0 (R5)	4
5,000 – 6,240	5,000 (R3)	0 (R5)	0 (R5)	5
6,250 – 7,490	5,000 (R3)	1,250 (R4)	0 (R5)	5
7,500 – 8,740	5,000 (R3)	2,500 (R3)	0 (R5)	6
8,750 – 8,990	7,500 (R2)	1,250 (R4)	0 (R5)	6
9,000 – 10,240	5,000 (R3)	0 (R5)	4,000 (R4)	7
10,250 – 11,490	5,000 (R3)	1,250 (R4)	4,000 (R4)	8
11,500 – 12,740	5,000 (R3)	2,500 (R3)	4,000 (R4)	8
12,750 – 13,990	7,500 (R2)	1,250 (R4)	4,000 (R4)	9
14,000 – 15,240	7,500 (R2)	2,500 (R3)	4,000 (R4)	9
15,250 – 16,740	5,000 (R3)	2,500 (R3)	8,000 (R3)	10
16,750 – 17,990	7,500 (R2)	1,250 (R4)	8,000 (R3)	10
18,000 – 19,240	7,500 (R2)	2,500 (R3)	8,000 (R3)	11
19,250 – 20,490	7,500 (R2)	3,750 (R2)	8,000 (R3)	11
20,500 – 21,740	10,000 (R1)	2,500 (R3)	8,000 (R3)	11
21,750 – 21,990	10,000 (R1)	3,750 (R2)	8,000 (R3)	11
22,000 – 23,240	7,500 (R2)	2,500 (R3)	12,000 (R2)	11
23,250 – 24,990	7,500 (R2)	3,750 (R2)	12,000 (R2)	11
24,500 – 25,740	7,500 (R2)	5,000 (R1)	12,000 (R2)	12
25,750 – 26,990	10,000 (R1)	3,750 (R2)	12,000 (R2)	12
27,000 – 30,990	10,000 (R1)	5,000 (R1)	12,000 (R2)	12
31,000 and above	10,000 (R1)	5,000 (R1)	16,000 (R1)	12

Table 8: Water Distribution

Available Water (kL/hour)	Hospital 1 Distribution (kL/hour)	Hospital 2 Distribution (kL/hour)	Hospital 3 Distribution (kL/hour)	Total Patient Output (Patients/hour)
0 – 59.9	0 (R5)	0 (R5)	0 (R5)	0
60 – 89.9	0 (R5)	60 (R4)	0 (R5)	1
90 – 149.9	90 (R4)	0 (R5)	0 (R5)	3
150 – 179.9	90 (R4)	60 (R4)	0 (R5)	4
180 – 239.9	180 (R3)	0 (R5)	0 (R5)	5
240 – 299.9	180 (R3)	60 (R4)	0 (R5)	5
300 – 329.9	180 (R3)	120 (R3)	0 (R5)	6
330 – 389.9	270 (R2)	60 (R4)	0 (R5)	6
390 – 449.9	270 (R2)	120 (R3)	0 (R5)	7
450 – 479.9	270 (R2)	180 (R2)	0 (R5)	7
480 – 509.9	180 (R3)	60 (R4)	240 (R4)	8
510 – 569.9	270 (R2)	0 (R5)	240 (R4)	8
570 – 659.9	270 (R2)	60 (R4)	240 (R4)	9
660 – 719.9	360 (R1)	60 (R4)	240 (R4)	9
720 – 779.9	360 (R1)	120 (R3)	240 (R4)	9
780 – 809.9	360 (R1)	180 (R2)	240 (R4)	10
810 – 869.9	270 (R2)	60 (R4)	480 (R3)	10
870 – 959.9	270 (R2)	120 (R3)	480 (R3)	11
960 – 1019.9	360 (R1)	120 (R3)	480 (R3)	11
1020 – 1079.9	360 (R1)	180 (R2)	480 (R3)	11
1080 – 1259.9	360 (R1)	240 (R1)	480 (R3)	11
1260 – 1319.9	360 (R1)	180 (R2)	720 (R2)	12
1320 – 1559.9	360 (R1)	240 (R1)	720 (R2)	12
1560 and above	360 (R1)	240 (R1)	960 (R1)	12

From the water and electricity distribution tables, we can obtain which rows the hospitals should operate given various amounts of water and electricity. But there are times of conflict, and they need to be resolved. There are three types of resource conflicts, which are explained.

Type 1: Insufficient Water

This type of conflict occurs when all three hospitals' water solutions are on rows below the electricity solutions. Water becomes the limiting resource and therefore the optimal solution would be the water solution. For example, there is a supply of 10,250 kW of electricity and 180 kL/hour of water. From the water distribution table, that provides Row 3 for Hospital 1, Row 5 for Hospital 2, and Row 5 for Hospital 3. From the electricity solution table, that provides Row 3 for Hospital 1, Row 4 for Hospital 2, and Row 4 for Hospital 3. Clearly, all of the hospital rows provided by the water solution are below the electricity solution. Therefore, the water solution is the optimal solution.

Type 2: Insufficient Electricity

This type of conflict occurs when all three hospitals' electricity solutions are on rows below the electricity solution. Electricity becomes the limiting resource and therefore the optimal solution would be the electricity solution. For example, the given supplies are 5000 kW of electricity and 810 kL/hour of water. According to the water solution table, the solution should be Row 2 for Hospital 1, Row 4 for Hospital 2 and Row 3 for Hospital 3. According to the electricity solution table, the solution should be Row 3 for Hospital 1, Row 5 for Hospital 2, Row 5 for Hospital 3. Clearly, all of the hospital rows provided by the electricity solution are below that of the water solution. Therefore, the electricity solution is the optimal solution.

Type 3: No Clear Insufficient Resource

In this type of conflict neither resource provides hospital rows that are all below the other resource. It is difficult to determine which resource is insufficient. For example, the supply given is 23,250 kW of electricity and 960 kL/hour of water. According to the electricity solution table, the solution is (Row 2, Row 2, Row 2) for Hospitals 1 to 3, respectively. According to the water solution table, the solution is (Row 1, Row 3, Row 3) for Hospitals 1 to 3, respectively. For Hospital 1, the electricity solution is below the water solution. But for Hospitals 2 and 3, the water solution is below the electricity solution. In this case, we would need to test out all the possibilities and rank them according to the discharge rates.

The first question to ask is how many choices are available? For Hospital 1 the choices are either Row 1 or Row 2. For Hospital 2 the choices are either Row 2 or Row 3. For Hospital 3 the choices are either Row 2 or Row 3. Therefore with three hospitals there are a total of 2^3, or 8 choices. The next step is to find out the patient output of each choice.

The possible choices are as follows:

(R1, R2, R2), (R2, R2, R2), (R1, R3, R2), (R1, R3, R3), (R2, R2, R3), (R2, R3, R2),
(R1, R2, R3), (R2, R3, R3)

Some choices would violate the resource constraints and can be eliminated. For example, if we choose (R1, R2, R2) that would consume 25,750 kW of electricity, and violates the electricity constraint. Similarly, other choices may violate water and/or electricity constraints. After eliminating five choices that violate resource constraints, there are three choices remaining: (R1, R3, R3), (R2, R2, R3) and (R2, R3, R3). Their patient output values are: (R1, R3, R3) = 10.92 or 11 patients/hour, (R2, R2, R3) = 10.83 or 11 patients/hour, (R2, R3, R3) = 10.58 or 11 patients/hour. Therefore the choice that provides the highest output, (R1, R3, R3) at 10.92 patients/hour is closest to 11 patients/hour. It is selected as the optimum solution for this combination of electricity and water input. The amount of electricity used is 20,500 kW and water used is 960 kL/hour. The amount of usage is within the constraints.

As can be seen from this scenario, when faced with the third type of conflict, the choices become exponential with more hospitals added. But it is possible to reduce the amount of calculation by eliminating choices that violate resource constraints.

There are three resource levels to evaluate this scenario: 1) 75% electricity and 25% water, 2) 25% electricity and 75% water and 3) 50% electricity and 50% water. The Matlab program is shown in Appendix A. The Matlab program first reads the available amounts of electricity and water from the I2Sim scenario, calculates the distribution ratios, and then sends them back. The process of assignment has been discussed. The I2Sim distributor requires percentages as distribution ratios. Equations 4.3 to 4.5 show how the distribution percentages are calculated.

$$Hospital\ i\ Electricity\ \% = \frac{Amount\ of\ elec\ assigned\ to\ Hospital\ i}{Total\ Electricity} \tag{4.3}$$

$$Water\ Station\ Electricity\ \% = \frac{Top\ Row\ of\ Water\ Station\ HRT}{Total\ Electricity} \tag{4.4}$$

$$Hospital\ i\ Water\ \% = \frac{Amount\ of\ water\ assigned\ to\ Hospital\ i}{Total\ Water} \tag{4.5}$$

4.1.1 75% Electricity and 25% Water

The scenario is first evaluated with 75% electricity and 25% water. There are 390 kL/hour of water and 24,000 kW of electricity available. The amounts of electricity and water are assumed to be constant throughout the 24-hour simulation period. Therefore, the ratios are only calculated once at the beginning of the simulation. Figure 16 shows the total number of discharged people at 160. The discharge figure may appear as a straight line, but in fact when zoomed in it should be a step function with each patient discharged. The water solution provides

Row 2, Row 3 and Row 5. The electricity solution provides Row 2, Row 2 and Row 2. The type of conflict is insufficient water. Therefore the water solution is followed. Table 9 shows the discharge rate for the hospitals. The electricity ratio calculated is [32.24% 16.12% 51.59% 0.04%]. The first three entries denote the amount of electricity distributed to Hospitals 1 to 3, respectively. The fourth entry denotes the electricity distributed to the water pump station. The water ratio calculated is [69.23% 30.77% 0.00%]. The three entries denote the amounts of electricity distributed to Hospitals 1 to 3, respectively.

Table 9: Patient Discharge Rates of the Hospitals

Hospital	Patient Discharge Rates (Patients/hour) **
1	6 (Row 2)
2	1 (Row 3)
3	0 (Row 5)

** Hospital 1 rounded from 5.7 to 6 patients per hour. Hospitals 2 and 3 without rounding.

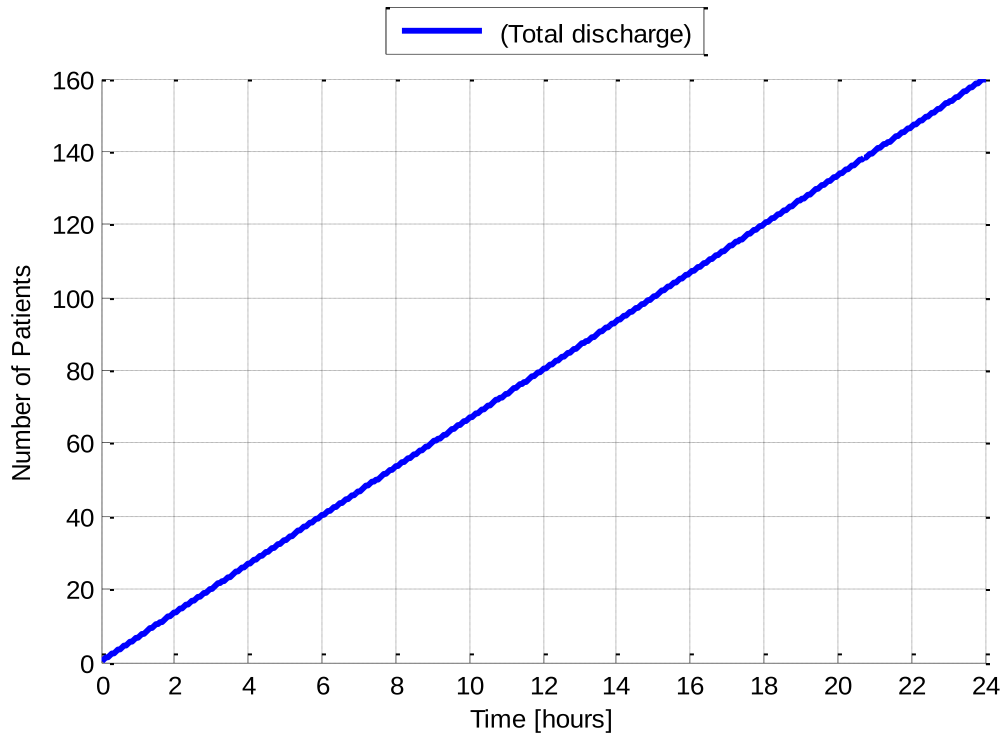

Figure 16: Total Number of Discharged Patients

4.1.2 25% Electricity and 75% Water

The scenario is evaluated with 25% electricity and 75% water. There are 8,000 kW of electricity and 1170 kL/hour of water available. Figure 17 shows the total number of people discharged is 138. The water solution provides Row 1, Row 1 and Row 3. The electricity solution provides Row 3, Row 3 and Row 5. Since electricity is the limiting factor, the solution follows electricity. Table 10 shows the discharge rates for the hospitals. The electricity ratio calculated is: [64.43% 32.22% 0% 0.13%]. The water ratio calculated is: [15.38% 10.26% 0%].

Table 10: Patient Discharge Rates of the Hospitals

Hospital	Patient Discharge Rates (Patients/hour) **
1	5 (Row 3)
2	1 (Row 3)
3	0 (Row 5)

** Hospital 1 rounded from 4.83 to 5 patients/hour. Hospitals 2 and 3 are without rounding.

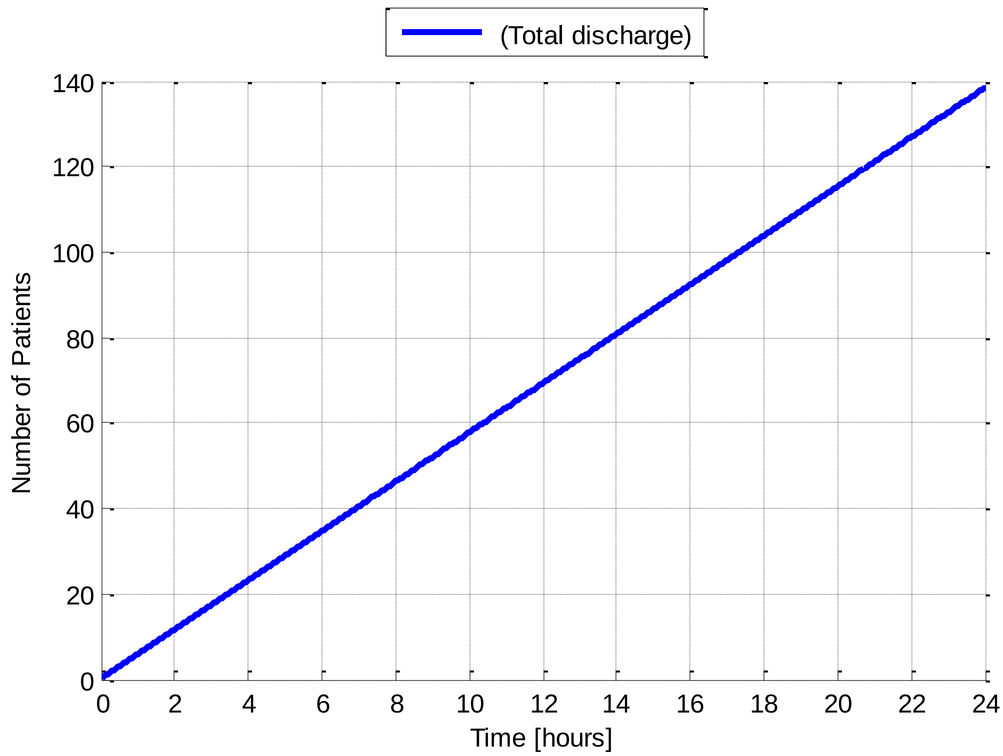

Figure 17: Total Number of Discharged Patients

4.1.3 50% Electricity and 50% Water

The scenario is next evaluated with 50% electricity and 50% water. There are 16,000 kW of electricity and 780 kL/hour of water available. The electricity solution provides Row 3, Row 3 and Row 3 for the three hospitals, respectively. The water solution provides Row 1, Row 2 and Row 4 for the three hospitals, respectively. Clearly, this is a type 3 conflict that requires listing of all the possibilities. After the calculation, the Matlab program selects Row 3, Row 3 and Row 3 as the optimal solution, because it provides the highest patient output without violating the resource limits. At this rate, the total discharge rate is 9.75 patients/hour. Figure 18 shows over a 24-hour period, there are 234 people discharged. Table 11 shows the discharge rates for the three hospitals. The electricity distribution calculated is [32.22% 16.11% 51.55% 0.06%]. The water distribution calculated is [23.08% 15.38% 61.54%].

Table 11: Patient Discharge Rates of the Hospitals

Hospital	Patient Discharge Rates (Patients/hour) **
1	5 (Row 3)
2	1 (Row 3)
3	4 (Row 3)

** Hospital 1 number is rounded from 4.83 to 5 patients/hour. Hospital 2 and 3 are without rounding.

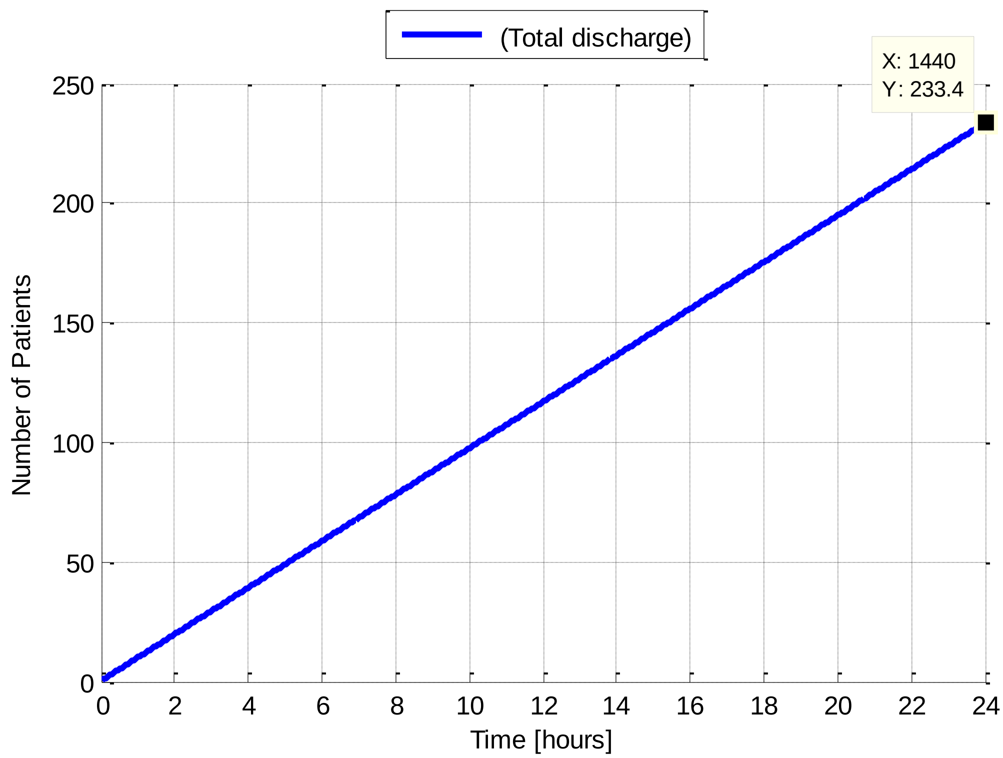

Figure 18: Total Number of Discharged Patients

4.2 Three Venues and Three Hospitals with Transportation with LBO Method

In the second scenario, the situation becomes more complicated. There are three venues and three hospitals involved. Instead of having a waiting room full of patients already at the hospitals, the patients originate from the venues and need to be transported to the hospitals. In addition to water and electricity, a third resource ambulance is added. Ambulance is an independent resource and does not depend on the other two resources. There are two objective functions in this scenario: 1) maximize the number of patients transported from the venues to the hospitals, 2) maximize the number of patients discharged from the hospitals. These two objective functions are independent and do not conflict with each other. Figure 19 shows the modeling scenario I built in I2Sim. It has the following descriptions.

- Each venue has 3,000 people inside initially.
- The incident at time zero is a riot. There are smoke and fire inside the venues. After the riot happens, 30% of the people are injured and 70% are healthy. The healthy are discharged from the venue.
- Someone threw a firebomb to the power lines supplying the electrical substation, which took out the lines (RM 5). But the substation itself is intact (PM 1). BC Hydro crew was sent to repair the damages. In this scenario, I simulated 100%, 50% and 0% resources.
- The three hospitals all require both water and electricity to operate.
- The capacities of the waiting rooms of the three hospitals are assumed to be very large (maximum 10,000 patients each). That would prevent any overflow of patients over the simulation period.
- The venues have standalone emergency supplies of water and electricity to enable people to exit safely and perform some simple on-site treatment.

- Each venue has three waiting areas where patients are loaded onto ambulances to the three hospitals.
- Each venue has three roads that lead to the three hospitals. There are 9 roads or channels in total.
- By transportation, it means the patients need to line up and wait at the venues for an ambulance to a hospital.
- The travel times of the ambulances from their station to the venues is neglected. Once the ambulances arrive at the venues, they travel back and forth between the venue and hospital.
- The Emergency Operations Centre (EOC), an entity not shown in the I2Sim scenario, has full authority over dispatch of ambulance, water and electricity.
- To maximize the number of patients transported to the three hospitals, the EOC needs to consider how to distribute the ambulances to the nine channels.
- To maximize the number of patients discharged from the hospitals, the EOC needs to consider how to distribute the water and electricity to the three hospitals.
- Each hospital has a small number of patients waiting inside already at time zero, to avoid the hospitals idling in the beginning.
- One key assumption is no distinction or triage on the granularity of the patients. That is, all patients are assumed to have sustained the same level of injury. All hospitals are assumed to perform the same operations to treat the patients, with differences in the levels of input and output.
- Venue 1 to Hospital 1 requires 10 minutes round trip with no traffic congestion.
- Venue 1 to Hospital 2 requires 20 minutes round trip with no traffic congestion.

- Venue 1 to Hospital 3 requires 30 minutes round trip with no traffic congestion.
- Venue 2 to Hospital 1 requires 20 minutes round trip with no traffic congestion.
- Venue 2 to Hospital 2 requires 30 minutes round trip with no traffic congestion.
- Venue 2 to Hospital 3 requires 10 minutes round trip with no traffic congestion.
- Venue 3 to Hospital 1 requires 30 minutes round trip with no traffic congestion.
- Venue 3 to Hospital 2 requires 10 minutes round trip with no traffic congestion.
- Venue 3 to Hospital 3 requires 20 minutes round trip with no traffic congestion.
- Each channel has its own HRT of ambulances required, number of people/hour and channel time. The roads to the hospitals have longer travel time if there are more ambulances on the road (i.e. congestion). The relationship amongst these quantities is shown in Equation 4.6. The inclusion of transportation is meant to make the scenario more realistic. This scenario is an excellent scenario to illustrate the LBO algorithm's ability to handle both countable and flow tokens.
- Each ambulance is assumed to carry one person per trip. Tables 12 to 20 show the HRTs of the 9 channels. Since transportation is independent objective, the rate of patient transport is given in per hour instead of per 12 hour.

$$Channel\ time = \frac{Number\ of\ Ambulances}{Patients/hour} \qquad (4.6)$$

To avoid hospitals idling at the beginning, each hospital is assumed to have 100 patients already in the waiting at time 0. Equation 4.7 shows the relationship between number of arrivals, discharged and waiting.

$$Total\ Arrival - Total\ Discharged - Initially\ Waiting = Total\ Waiting \qquad (4.7)$$

This scenario is evaluated with three resource levels listed below.

- 50% Electricity, 50% Water and 50% Ambulances
- 25% Electricity, 50% Water and 75% Ambulances
- 50% Electricity, 25% Water and 75% Ambulances

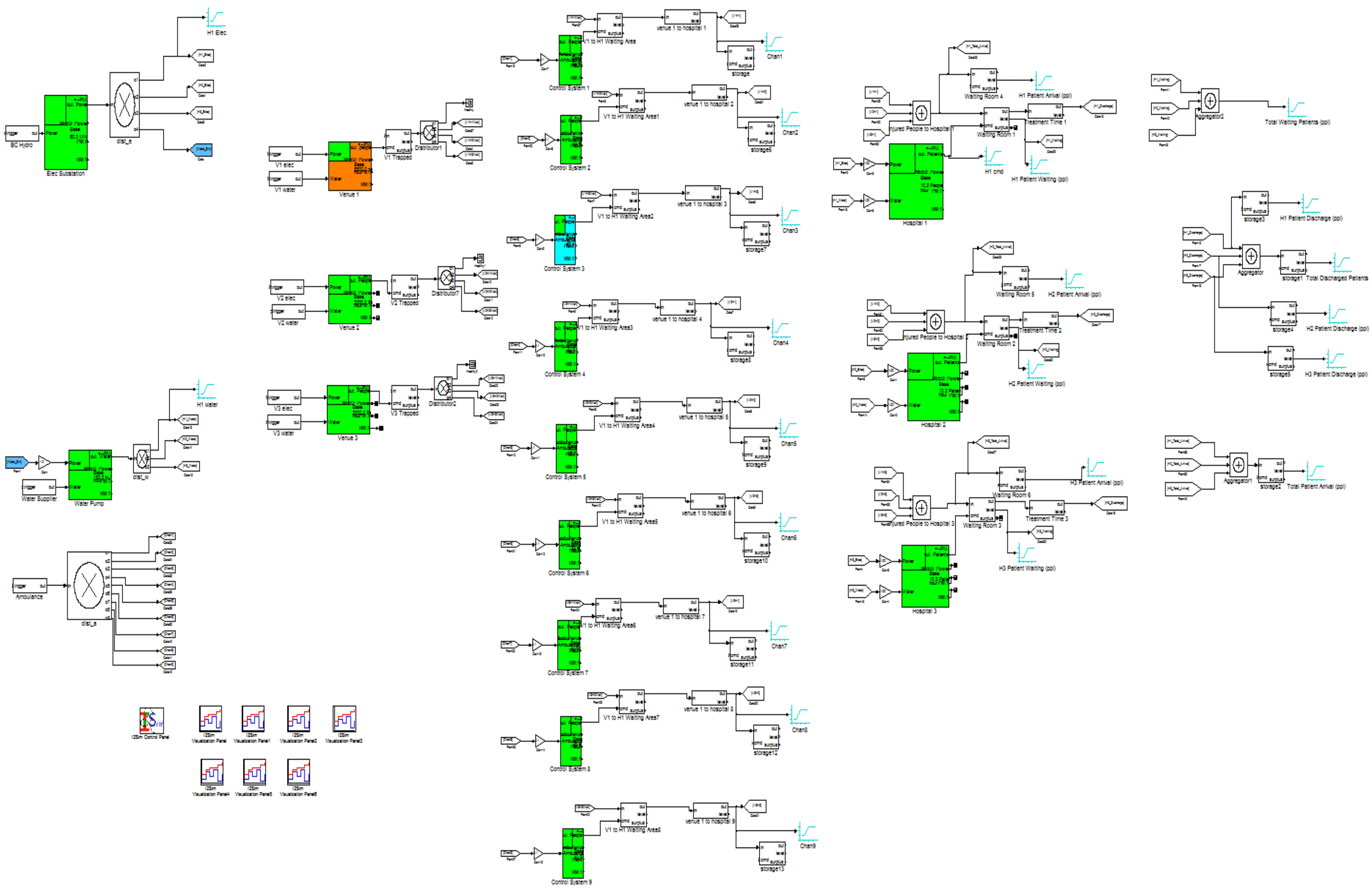

Figure 19: I2Sim Three Venues and Three Hospitals with Transportation Scenario

Table 12: Venue 1 to Hospital 1 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
16	53	18
12	48	15
8	40	12
4	24	10
0	0	-

Table 13: Venue 1 to Hospital 2 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
8	17	28
6	14	25
4	11	22
2	6	20
0	0	-

Table 14: Venue 1 to Hospital 3 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
16	22	44
12	19	37
8	14	33
4	8	30
0	0	-

Table 15: Venue 2 to Hospital 1 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
16	32	30
12	29	25
8	22	22
4	12	20
0	0	-

Table 16: Venue 2 to Hospital 2 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
8	12	40
6	10	37
4	7	33
2	4	30
0	0	-

Table 17: Venue 2 to Hospital 3 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
16	53	18
12	48	15
8	40	12
4	24	10
0	0	-

Table 18: Venue 3 to Hospital 1 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
16	22	44
12	19	37
8	14	33
4	8	30
0	0	-

Table 19: Venue 3 to Hospital 2 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
8	26	18
6	24	15
4	20	12
2	12	10
0	0	-

Table 20: Venue 3 to Hospital 3 HRT

Number of Ambulances	Patients/hour	Channel time (round trip, min)
16	32	30
12	29	25
8	22	22
4	12	20
0	0	-

Tables 21 to 25 show the HRTs for the three hospitals, water station and electrical substation. The data is taken from a confidential Olympics report and modified. They show the best Vancouver-area hospitals with a maximum discharge capacity of 10 patients/hour.

Table 21: Hospital 1 HRT

Patient Output (Patients/hour) **	Electricity Input (kW)	Water Input (kL/hour)
10	10,000	51
10	7,500	38
8	5,000	26
5	2,500	13
0	0	0

** Actual patient outputs are (patients/hour): 10, 9.6, 8, 5 and 0.

Table 22: Hospital 2 HRT

Patient Output (Patients/hour) **	Electricity (kW)	Water (kL/hour)
10	20,000	61
10	15,000	46
7	9,000	31
3	3,000	15
0	0	0

** Actual patient outputs are (patients/hour): 10, 9.2, 7, 3 and 0.

Table 23: Hospital 3 HRT

Patient Output (Patients/hour) **	Electricity (kW)	Water (kL/hour)
10	30,000	71
10	23,000	53
8	15,000	36
5	8,000	18
0	0	0

** Actual outputs are (patients/hour): 10, 9.5, 8, 5 and 0.

Table 24: Water Pump Station HRT

Water (kL/hour)	Electricity (kW)	Water (kL/12 hours)
183	10	183
138	7.5	138
93	5	93
46	2.5	46
0	0	0

Table 25: Electrical Substation HRT

Output Power (kW)	Input Power (kW)
61,000	61,000
47,000	47,000
31,000	31,000
16,000	16,000
0	0

Tables 26 to 28 show the partial distributions for ambulance, water and electricity, respectively.

Table 26: Partial Ambulance Distribution

# of Amb.	Chan. 1	Chan 2	Chan. 3	Chan. 4	Chan. 5	Chan. 6	Chan. 7	Chan. 8	Chan. 9	Total Patients/ hour
144	16	8	16	16	8	16	16	8	16	269
108	16	8	12	12	8	16	12	8	16	262
72	12	6	4	12	4	12	4	6	12	215
…	…	…	…	…	…	…	…	…	…	…
0	0	0	0	0	0	0	0	0	0	0

Table 27: Partial Water Distribution

Water Available (kL/hour)	Hospital 1 Distribution (kL/hour)	Hospital 2 Distribution (kL/hour)	Hospital 3 Distribution (kL/hour)	Patient Output (Patients/hour)
183	51	61	71	31
…	…	…	…	…
93	26	31	36	23
…	…	…	…	…
0	0	0	0	0

Table 28: Partial Electricity Distribution

Electricity Available (kW)	Hospital 1 Distribution (kW)	Hospital 2 Distribution (kW)	Hospital 3 Distribution (kW)	Water Station Distribution (kW)	Patient Output (Patients/hour)
61,000	10,000	20,000	30,000	10	31
…	…	…	…	…	…
31,000	5,000	9,000	15,000	10	23
…	…	…	…	…	…
0	0	0	0	0	0

4.2.1 50% Electricity, 50% Water and 50% Ambulances

The scenario is performed with 50% electricity, 50% water and 50% ambulances. There are 31,000 kW of electricity, 93 kL/hour of water and 72 ambulances available. The electrical distribution calculated is [24.99%, 29.99%, 26.66%, 0.03%]. The first percentage denotes amount of electricity dispatched to Hospital 1. The second and third percentages denote the amounts to Hospitals 2 and 3, respectively. The last percentage denotes the amount sent to the water pump station. The water distribution calculated is [40.86%, 33.33%, 19.36%]. The three percentages denotes the amounts sent to Hospitals 1, 2 and 3, respectively. The ambulance distribution calculated is [16.67%, 8.33%, 5.56%, 16.67%, 5.56%, 16.67%, 5.56%, 8.33%, 16.67%]. The percentages denote the 9 channels, starting with Venue 1 to Hospital 1. Hospital 1 operates at Row 2 of its HRT. Hospital 2 operates at Row 3. Hospital 3 operates at Row 4. The total discharge rate is 22 patients/hour.

Figure 20 shows the total discharged, arrived and waiting patients for the three hospitals together. The total number of discharged patients is 519. The total number of arrived patients is 2700. The total number of waiting patients is 2460. The initially number of waiting patients is 279. The number of waiting patients reaches a maximum around 15 hours and declines after that.

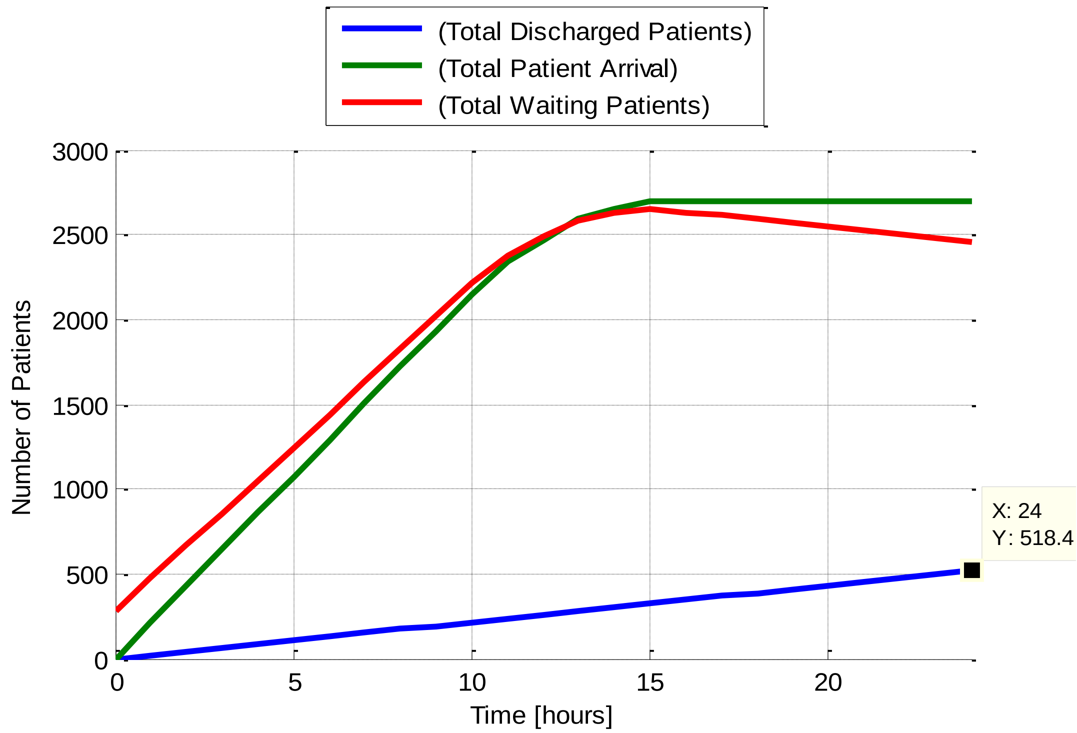

Figure 20: Total Number of Discharged, Arrival and Waiting Patients

Figure 21 shows the total discharged, arrived and waiting patients for Hospital 1. The total number of discharged patients is 231. The total number of arrived patients is 1040. The total number of waiting patients is 900. The number of initially waiting patients is 91.

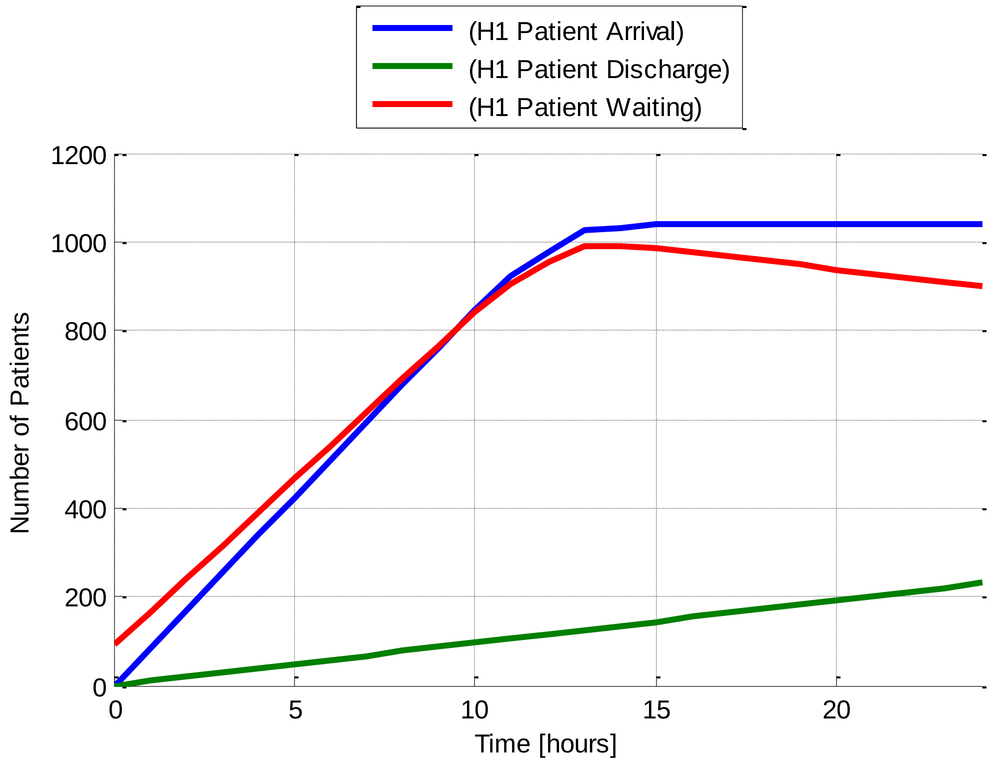

Figure 21: Hospital 1 Number of Arriving, Waiting and Discharged Patients

Figure 22 shows the total arrived, waiting and discharged patients for Hospital 2. The total number of discharged patients is 168. The total number of arrived patients is 618. The total number of waiting patients is 543. The initially number of waiting patients is 93.

Figure 23 shows the total arrived, waiting and discharged patients for Hospital 3. The total number of discharged patients is 120. The total number of arrived patients is 1,042. The total number of waiting patients is 1,017. The number of initially waiting patients is 95.

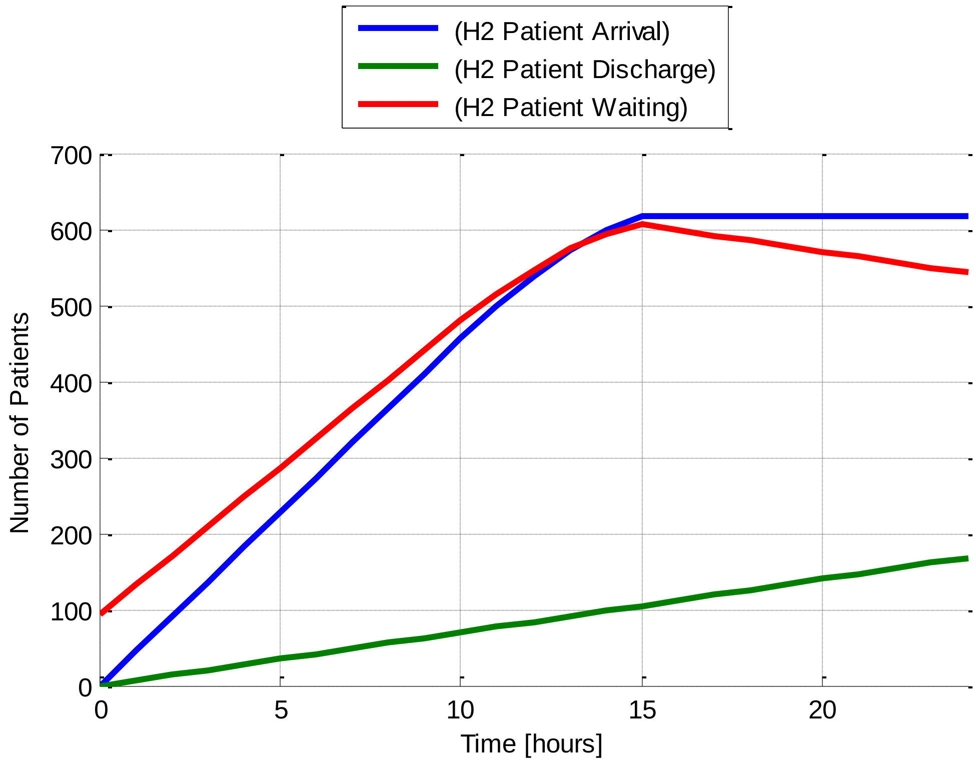

Figure 22: Hospital 2 Number of Arriving, Waiting and Discharged Patients

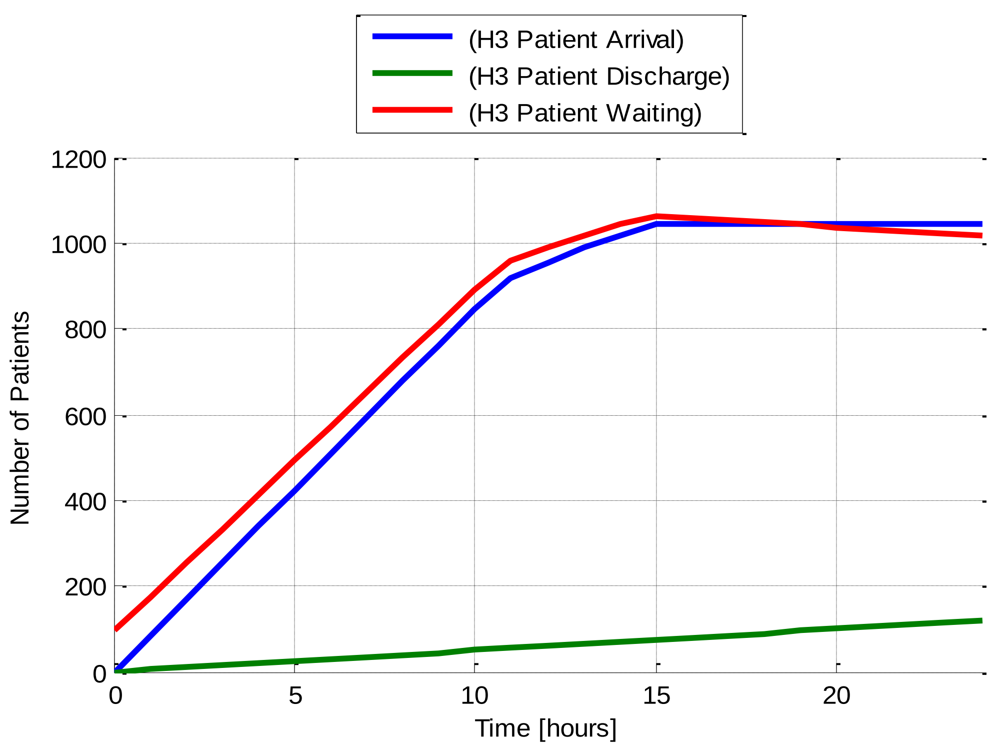

Figure 23: Hospital 3 Number of Arriving, Waiting and Discharged Patients

4.2.2 25% Electricity, 50% Water and 75% Ambulances

The scenario is evaluated with 25% electricity, 50% water and 75% ambulance. There are 16,000 kW of power, 93 kL/hour of water and 108 ambulances available. The electricity ratio calculated is: [33.31% 59.96% 0% 0.07%]. The water ratio calculated is: [27.96% 33.33% 0%]. The ambulance ratio calculated is: [14.82% 7.41% 11.11% 11.11% 7.41% 14.82% 11.11% 7.41% 14.82%]. Hospital 1 operates in Row 3 of its HRT. Hospital 2 operates in Row 3 of its HRT. Hospital 3 operates in Row 5 of its HRT.

Please notice since Hospital 3 is not given any resources, there should not be any patients sent to that hospital. Only 6 channels have ambulance running in them. The extra ambulances should be distributed to the remaining channels to help transport patients. Therefore, the ambulance ratios need to be adjusted in order to maximize the number of people treated. The channel from Venue 2 to Hospital 1 is given 4 more ambulances, boosting it to 16 ambulances, or the first row in its HRT. The channel from Venue 3 to Hospital 1 is given 4 more ambulances, boosting it to 16 ambulances, also the first row in its HRT. When the six channels are filled to their top rows, there are still 36 ambulances undistributed. These ambulances should be given back to the EOC and dispatched elsewhere. After the redistribution, the ambulance ratio is: [22.22% 11.11% 0% 22.22% 11.11% 0% 22.22% 11.11% 0%]. The electricity and water ratios are not affected.

Figure 24 shows the total number of arrival, waiting and discharged patients. Over the 24-hour period, there are 365 people discharged from the hospitals. The total number of arrived patients is 2700. The total number of waiting patients is 2666, which occurs at 21 hours, and it declines after that. The number of initially waiting patients is 279. Figures 25 and 26 show the arrived, waiting and discharged patients for Hospitals 1 and 2, respectively. Hospital 3 is not shown since it is not operating.

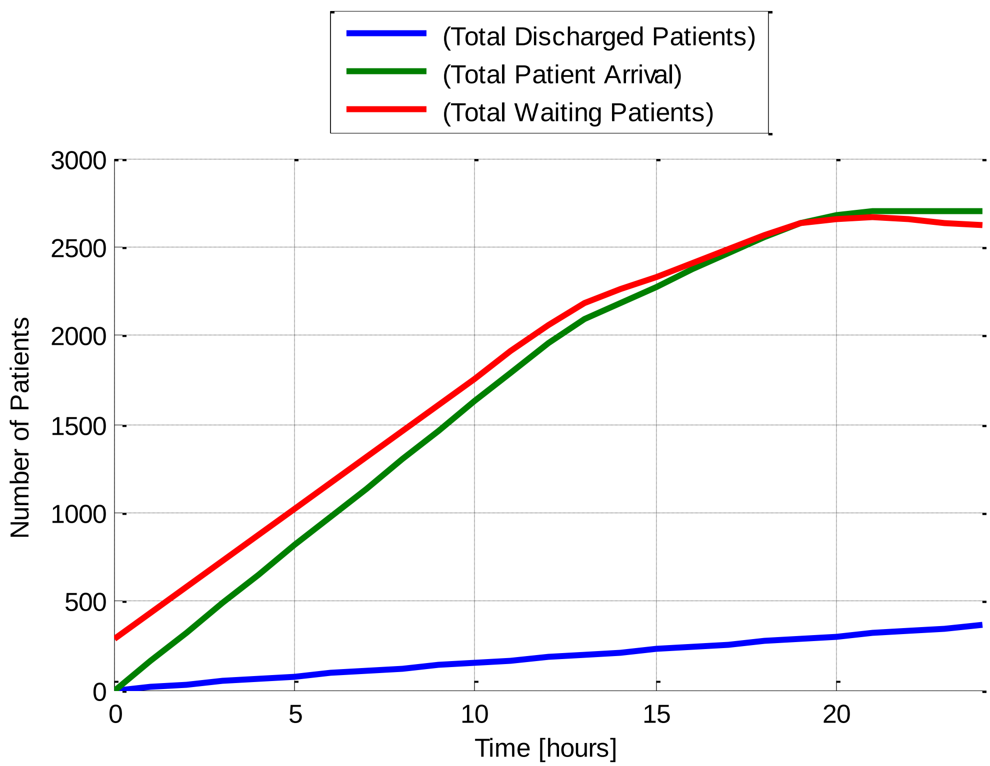

Figure 24: Total Number of Discharged, Arrival and Waiting Patients

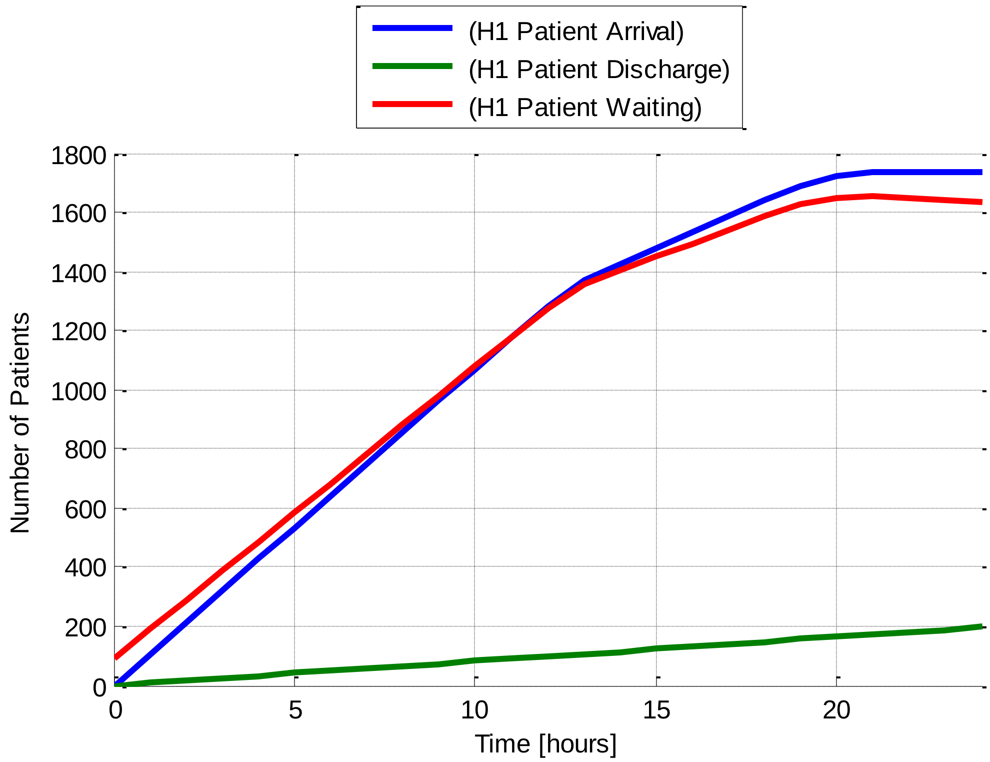

Figure 25: Hospital 1 Number of Arriving, Waiting and Discharged Patients

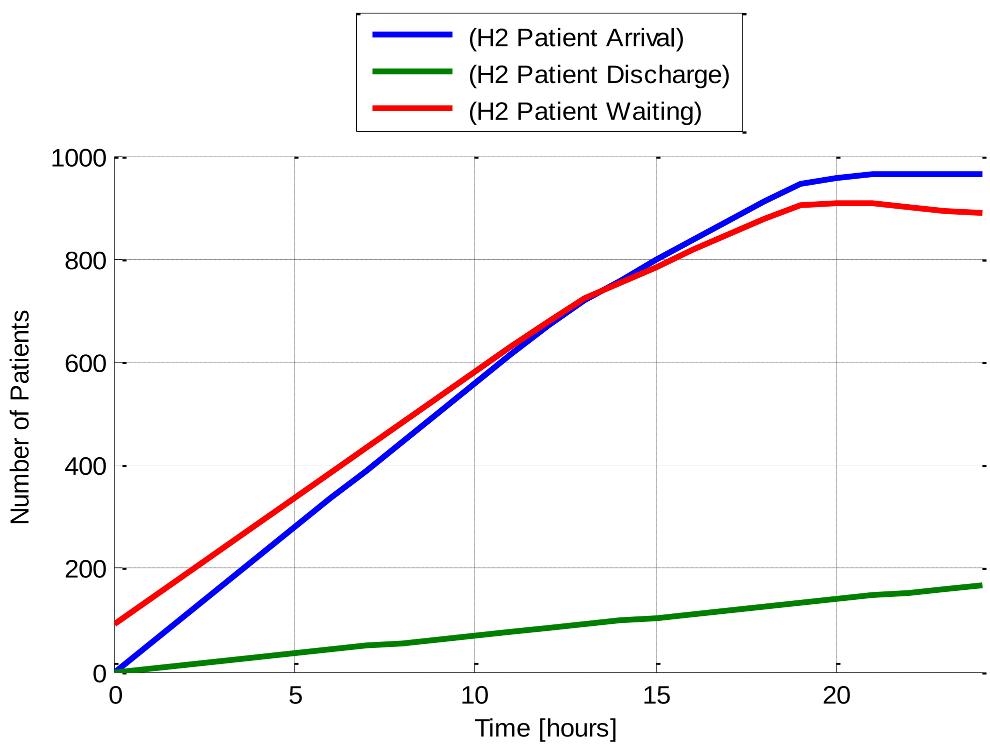

Figure 26: Hospital 2 Number of Arriving, Waiting and Discharged Patients

4.2.3 50% Electricity, 25% Water and 75% Ambulances

The scenario is tested with 50% electricity, 25% water and 75% ambulances. There are 31,000 kW of power, 46 kL/hour of water and 108 ambulances available. The electricity ratio calculated is: [16.66% 0% 26.66% 0.03%]. The water ratio calculated is: [56.52% 0% 39.13%]. The ambulance ratio calculated is: [14.82% 7.41% 11.11% 11.11%7.41% 14.82% 11.11% 7.41% 14.82%]. Hospital 1 operates in Row 3 of its HRT. Hospital 2 operates in Row 5 of its HRT. Hospital 3 operates in Row 3 of its HRT.

Since Hospital 2 is not operating, the ambulances need to be redistributed. There are only 6 channels running. The channel from Venue 1 to Hospital 3 is given 4 more ambulances. The same also applies for the Venue 2 to Hospital 1 channel and the Venue 3 to Hospital 2 channel. There are a total of 96 ambulances operating. The remaining ambulances are given back to the EOC. After the redistribution, the ambulance ratio is [16.67% 0% 16.67% 16.67% 0% 16.67% 16.67% 0% 16.67%]. The electricity and water ratios are not affected.

Figure 27 shows the total number of discharged, arrived and waiting patients. Over the 24-hour period, there are 317 people discharged. The number of arrived patients is 2700. The maximum number of waiting patients is 2762, which occurs at 17 hours, and then it declines. Figures 28 and 29 show the number of discharged, arrived and waiting patients for Hospitals 1 and 3, respectively.

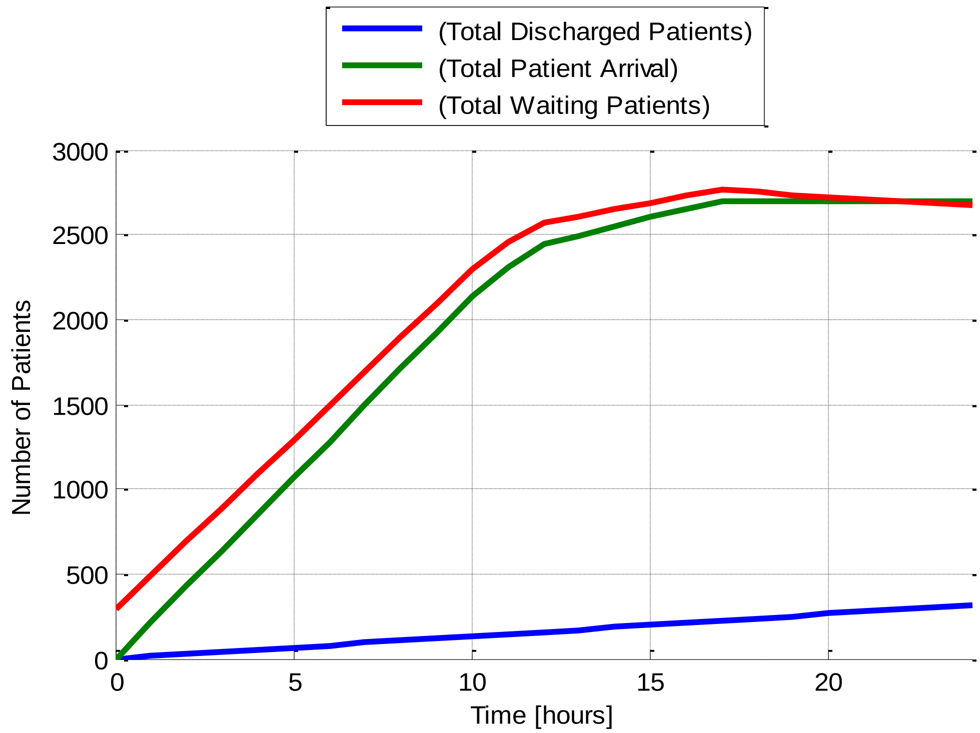

Figure 27: Total Discharged, Arrival and Waiting Patients

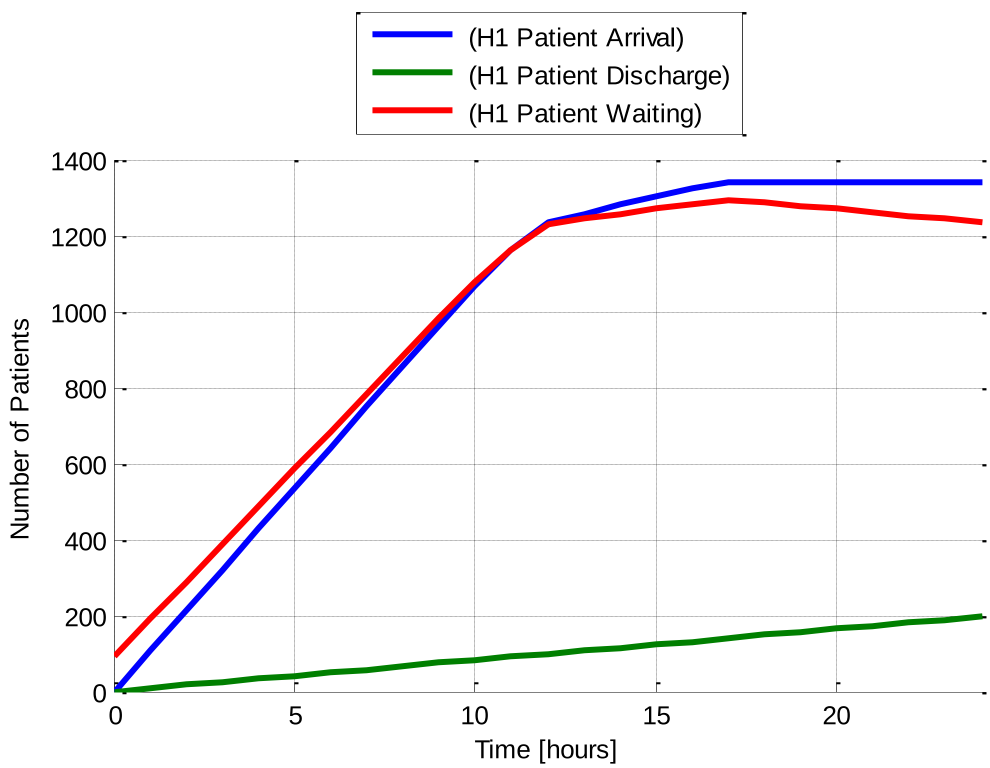

Figure 28: Hospital 1 Arriving, Waiting and Discharged Patients

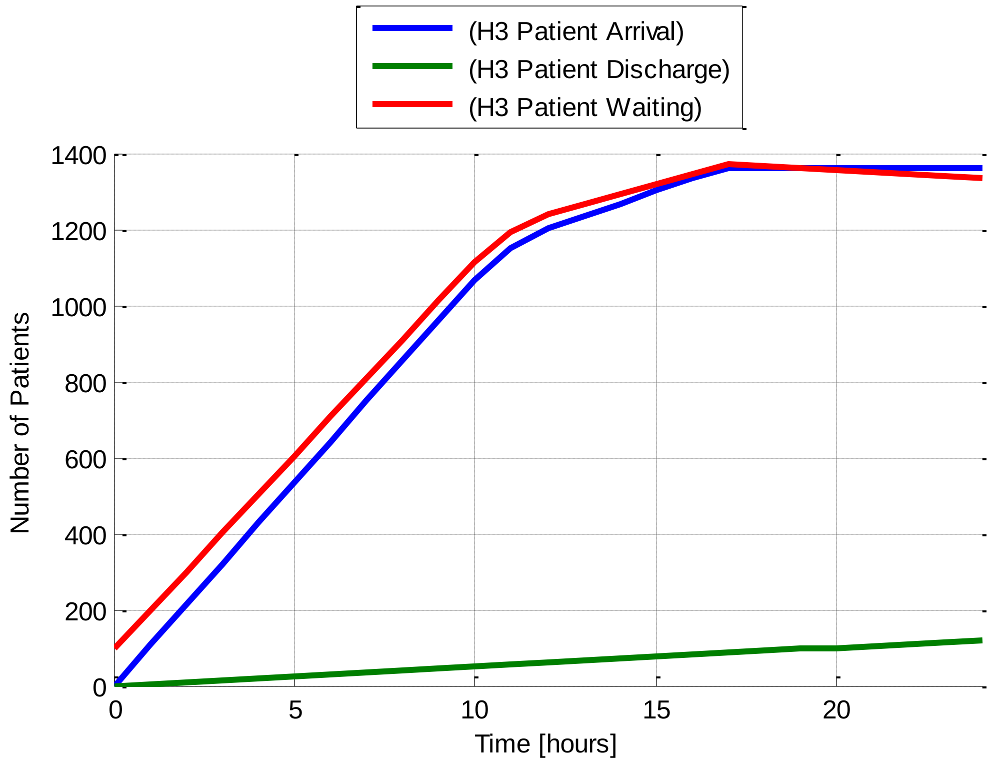

Figure 29: Hospital 3 Arriving, Waiting and Discharged Patients

4.3 Three Hospitals No Transportation Scenario with RL Method

In this section, the scenario mentioned in section 4.1 is evaluated again for comparison. This time the scenario is tested with the RL method. The modeling scenario setup in I2Sim that I built is shown in Figure 30. Instead of having an external Matlab program to read the inputs, the DAARTS agent (block with the robot picture) is placed inside I2Sim with the other blocks. DAARTS agent has 12 different inputs. They are the PMs and RMs of the three hospitals, electrical substation and water station (10 inputs), the total discharge rate of the three hospitals (reward) and the total waiting room level. The outputs produced are the distribution ratios for water and electricity.

Tables 29 and 30 show the combinations for electricity and water distributions that are available to the agent. Some of the ratios are chosen by the Lagrange method (Combinations 1 and 2 in Table 29; Combinations 1, 2 and 3 in Table 30). Others are educated guesses. The terminated column denotes any undistributed quantity. In discussions with Khouj, it is important that each of the ratio maps to a different row in the HRT.

If the water pump station is not receiving enough electricity to supply the hospitals, then the outputs are affected. In that case, there must be some emergency water storage inside the hospital to bring the service up to a certain level.

Table 29: Electricity Distribution Combinations

Combination	Hospital 1 (%)	Hospital 2 (%)	Hospital 3 (%)	Water Pump Station (%)	Terminated (%)	Total (%)
1	32.25	16.12	51.60	0.03	0	100
2	64.43	32.22	0	0.13	3.22	100
3	33	33	33	1	0	100
4	50	0	49	0.06	0.94	100
5	10	40	49.95	0.05	0	100

Table 30: Water Distribution Combinations

Combination	Hospital 1 (%)	Hospital 2 (%)	Hospital 3 (%)	Terminated (%)	Total (%)
1	23.08	15.38	61.54	0	100
2	15.38	10.26	0	74.36	100
3	69.23	30.77	0	0	100
4	33.33	33.33	33.33	0.01	100
5	10	40	50	0	100

As mentioned before, there are five production cells which send their Physical Modes and Resource Modes to DAARTS: electrical substation, water pump station, Hospital 1, Hospital 2 and Hospital 3. Therefore, there are 5^5 = 3125 states. The number of actions under each state is 5 power combinations * 5 water combinations = 25 actions. The number of Physical Mode is one. Therefore, the Look Up Table has a size of 3,125 * 25 * 1 = 78,125.

The size of the LUT can grow exponentially. If there are 10 production cells, that would provide 5^{10} = 9,765,625 states! The size of the LUT becomes 5^{10} states * 25 actions * 1 PM = 244,140,625! Clearly there needs to be a way of approximating the LUT. In Khouj's PhD proposal, he mentioned Artificial Neural Networks (ANN) as a way of approximating the LUT [24].

The simulation time is over 24 hours with one minute time steps. The Matlab code for the agent can be found in Appendix C. It is an adapted version from Khouj's PhD proposal [24], used with permission. The entries in the LUT are initialized to random numbers on the first run. Then the agent learns how to handle the situation in each time step. Each time step the agent senses a state, selects an action that has the highest Q-value and receives a reward. If the agent receives a positive reward, he will keep performing the same action. If the agent receives a negative reward, he will try another action. This scenario is tested with 100%, 50% and 0% resource levels. The following cases are tested.

- 75% Electricity and 25% Water
- 25% Electricity and 75% Water
- 50% Electricity and 50% Water

The learning parameters for all the cases are listed below. The learning rate and discount factor are optimal combinations from Khouj's PhD proposal [24]. The exploratory rate is set as every 144 steps.

- $\alpha = 0.5$
- $\gamma = 0.7$
- ε = every 144 steps

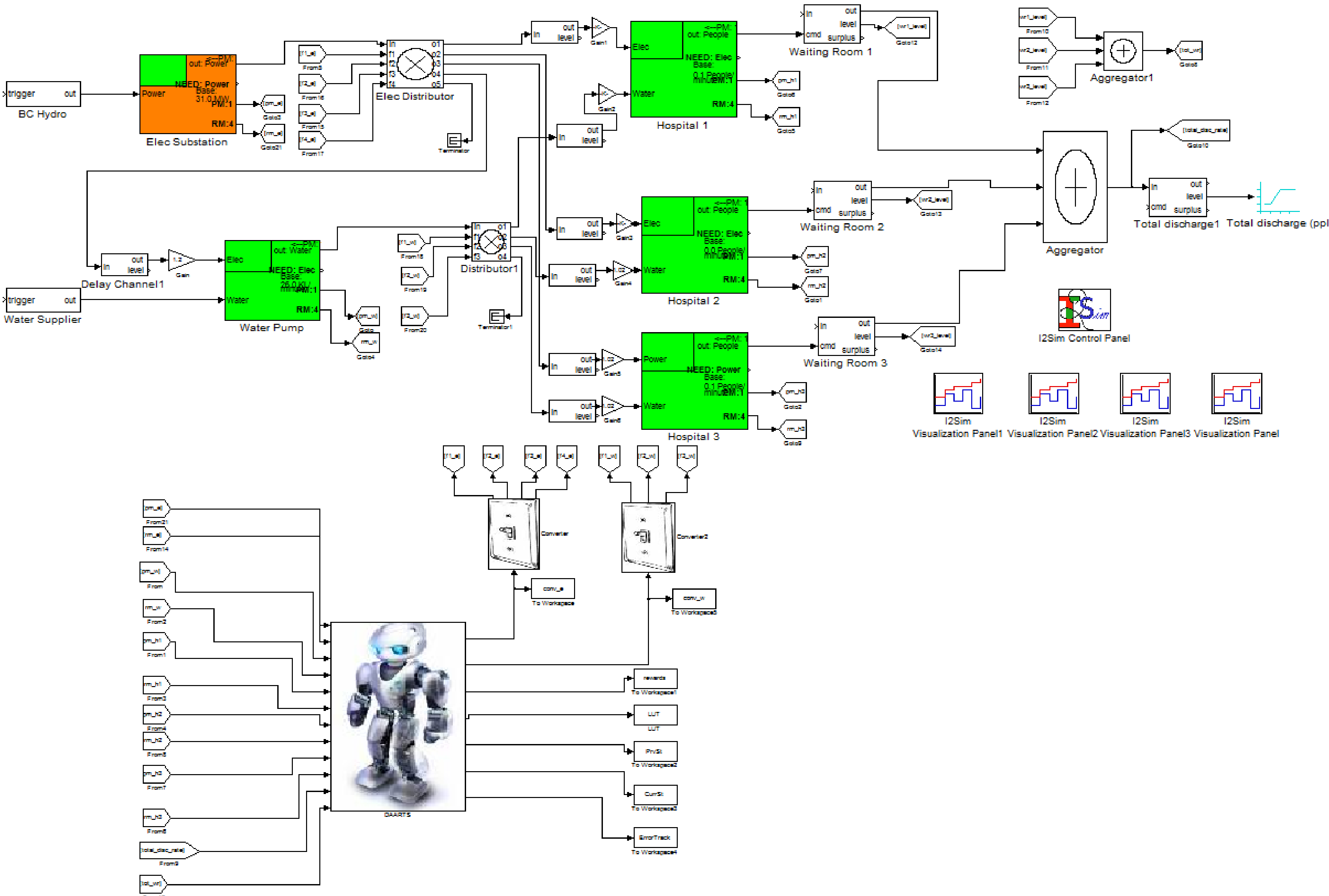

Figure 30: I2Sim Scenario Three Hospitals with No Transportation with DAARTS

4.3.1 75% Electricity and 25% Water

The scenario is tested with 75% electricity and 25% water. There are 24,000 kW of electricity and 390 kL/hour of water available. Figure 31 shows the total number of discharged people at 156 after 67 runs. The electricity ratio selected by the agent is: [32.24% 16.12% 51.60% 0.03%]. The water ratio selected by the agent is: [69.23% 30.77% 0%]. Hospital 1 operates in Row 2 of its HRT. Hospital 2 operates in Row 3 of its HRT. Hospital 3 operates in Row 5 of its HRT.

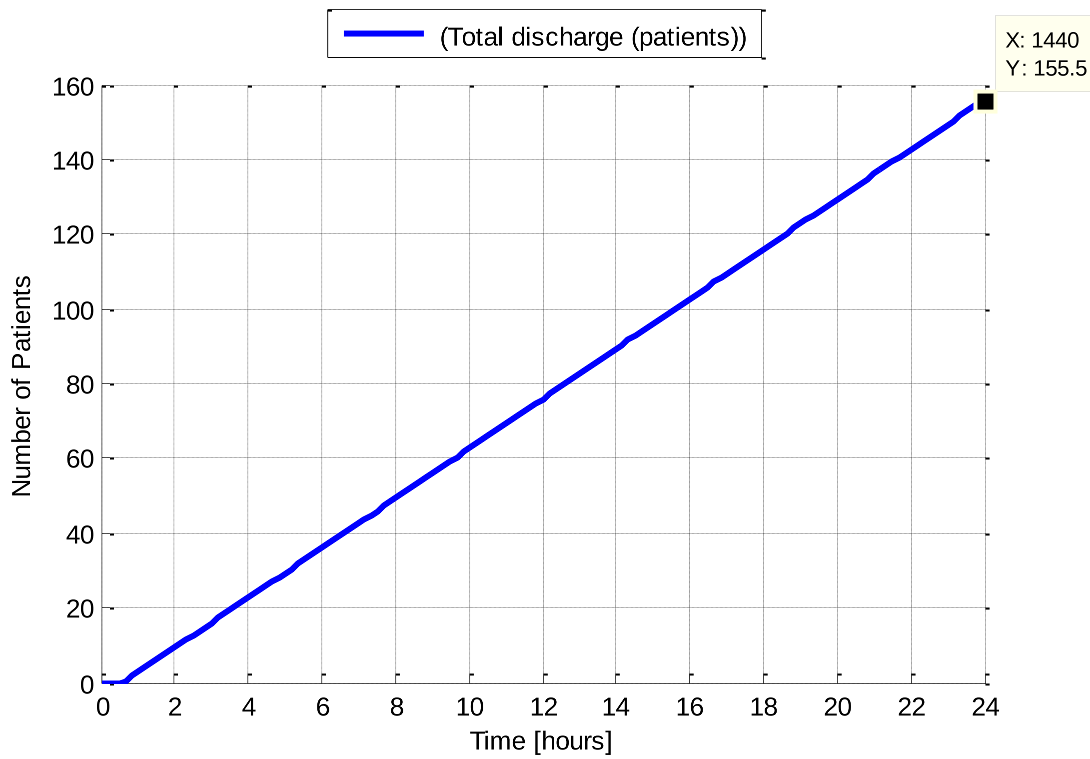

Figure 31: Total Number of Discharged Patients

4.3.2 25% Electricity and 75% Water

The scenario is tested with 25% electricity and 75% water. There are 8,000 kW of electricity and 1170 kL/hour of water available. Figure 32 shows the total number of discharged people is 138 after 39 runs. This matches with the LBO method. The electricity ratio selected by the agent is [64.43% 32.22% 0% 0.13%]. The water ratio selected by the agent is [15.38% 10.26%

0%]. Hospital 1 operates in Row 3 of its HRT. Hospital 2 operates in Row 3 of its HRT. Hospital 3 operates in Row 5 of its HRT.

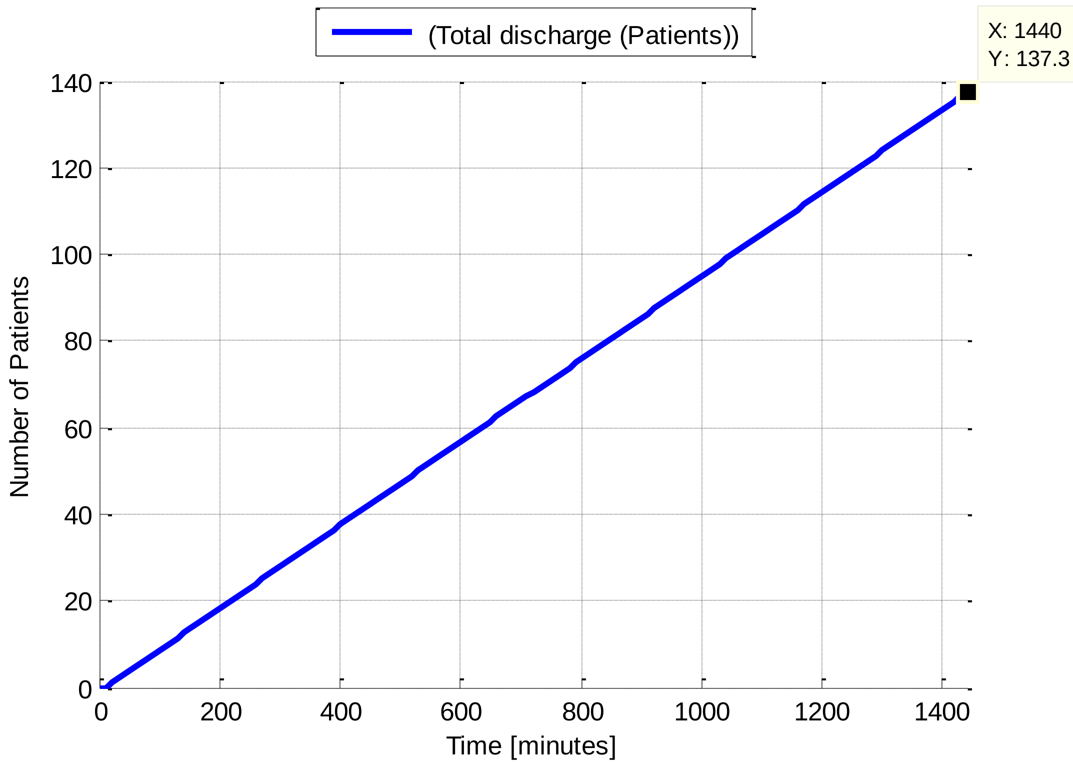

Figure 32: Total Number of Discharged Patients

4.3.3 50% Electricity and 50% Water

The scenario is tested with 50% electricity and 50% water. There are 16,000 kW of electricity and 780 kL/hour of water available. Figure 33 shows the total number of discharged patients is 231 after 16 runs, which matches with the expected number. The electricity distribution ratio selected by the agent is [32.25% 16.12% 51.60% 0.03%]. The water distribution ratio selected by the agent is [23.08% 15.38% 61.54%]. Hospital 1 operates in Row 3 of its HRT. Hospital 2 operates in Row 3 of its HRT. Hospital 3 operates in Row 3 of its HRT.

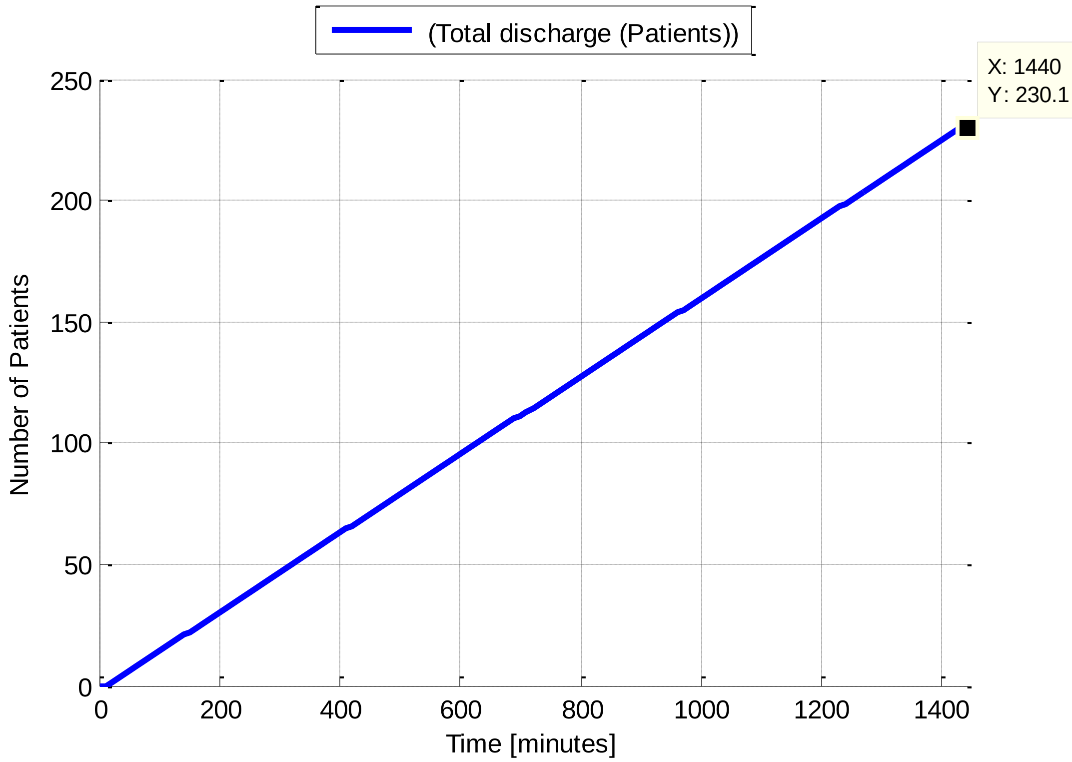

Figure 33: Total Number of Discharged Patients

4.4 Discussion of the Results and Comparison of the Two Methods

This section discusses the comparison results of the Three Hospitals with No Transportation scenario. By having a comparison, we can better understand the strengths and weaknesses of each method. Generally speaking, the Lagrange method is faster with a narrower focus, whereas the RL method is more adaptable. RL method is capable of distributing to both venues and hospitals.

Table 31 shows comparisons of LBO and RL methods for three different cases. The two methods produce matching results. Table 32 shows the inputs required and outputs produced by both methods. When there are 100% electricity and 100% water, the number of patients discharged is 288 over a 24-hour period. Each run of the LBO method takes about 5 seconds.

Each run of the RL method takes about 5 minutes. While the time difference may not be a fair comparison, but it still shows that LBO has very fast calculation speed. I think the two methods can complement each other. One can use the Lagrange Based Optimization method to calculate some educated choices for the RL agent, and run the agent to find out the optimum results.

Table 31: Comparison of Three Hospitals with No Transportation Results

	Lagrange Based Optimization Method	**Reinforcement Learning Method**
75% Electricity and 25% Water	160 people discharged	156 people discharged (after 67 runs)
25% Electricity and 75% Water	138 people discharged	138 people discharged (after 39 runs)
50% Electricity and 50% Water	234 people discharged	231 people discharged (after 16 runs)

Table 32: Inputs Required and Outputs Produced

	Inputs Required	Outputs Produced
Lagrange Based Optimization Method	• HRTs of hospitals, • Amount of water and electricity available	• Calculated optimal distribution ratios of water and electricity
Reinforcement Learning Method	• PM and RM of production cells • Learning rate (α) • Discount factor (γ) • Exploration rate (ε) • Pre-defined set of distribution ratios	• Selected optimal distribution ratios from pre-defined choices

For the three venues and three hospitals, the LBO algorithm was able to calculate the optimal distribution ratios for the ambulances, water and electricity. But the ambulance distribution assumes all the hospitals are operating. When one of the hospitals is shut down due to insufficient resources, I had to manually redistribute the ambulance ratios. This is one of the future improvements that can be made.

For the RL method, the choice of the learning parameters and the pre-defined distribution choices affect the results. The more pre-defined choices an agent is given, the more probable he is able to make the optimal decision. But the trade-off is longer time to explore all of the actions. It is a very time consuming process to experiment with different sets of parameters to find the optimal one.

Table 33 shows comparison of the two methods with a list of four criteria. Table 34 lists the advantages and disadvantages of each method. Generally speaking, the two methods can be used to complement each other. One can use Lagrange to calculate an educated choice for the agent, and run the agent to find the optimal results.

Table 33: Comparison of Two Methods

	Accuracy of Results	Robustness	Speed	Scalability
Lagrange Based Optimization Method	Very good	Good	Very fast.	Good scalability. Number of hospitals and resources can increase with small increase in computation time.
Reinforcement Learning Method	Good	Good	Relatively slower as the agent requires time to learn and explore all the states.	The state space may become large quickly; need a way of approximating the LUT.

Table 34: Advantages and Disadvantages of Both Methods

	Advantages	Disadvantages
Lagrange Based Optimization Method	• Fast calculation • Accurate results • Simple algorithm	• Still needs some modifications if a distributor is to give to venue and hospital at same time • Does not guarantee an answer will be found in every case, but if one is found it will be the optimum • Requires detailed information such as knowledge of the full HRTs • Worst case scenario of conflict, the algorithm could degenerate into 2^n time complexity, where n is the number of hospitals
Reinforcement Learning Method	• Very adaptable to different situations • Does not require full knowledge of the production cells (i.e. full HRTs)	• Time consuming for the agent to find the optimal solution • Size of the LUT grows exponentially with additional productions cells, actions and PMs • Agent's actions need to be carefully defined to ensure each one maps to a specific row in the HRT.

As discussed previously, in a real situation, the power supplied to a hospital cannot be easily controlled. Generally there are no controls in place to reduce power consumption to a hospital in a controlled manner. Traditionally the only ways the supplied power can be reduced are [48]:

1. Load shedding: Load shedding of selected loads by turning on/off the feeders from a substation, providing 0% or 100% load.

2. Reducing the feeder voltage: Lowering the line voltage may reduce the system load but it is limited in the achievable load reduction and for many modern electronic loads that use switching power supplies there is no net load reduction.

3. Voluntary load shedding: The hospital could be contacted and requested to reduce their load by turning off non-essential appliances.

With the move to a smart grid it will be possible to implement much greater control on the power system load [48]. In the future smart grid and smart metering technologies can provide automatic smart load shedding. It will become possible to auto detect and monitor the load conditions and to remotely shut down the non-critical loads. In addition with communication to the loads not only could the loads be turned off it could also become possible to change settings. For example the temperature setting of heating and air conditioning could be changes to reduce the load. This could also help to reduce load in emergency situations so that the available power would be used at the optimal location.

5 Conclusions and Future Work

The proposed Lagrange Based Optimization algorithm shows very good suitability for use in I2Sim. In the first scenario of the three hospitals no transportation, the LBO algorithm is able to accurately calculate the optimum water and electricity distribution ratios for the three different resource levels (75% electricity and 25% water; 25% electricity and 75% water; 50% electricity and 50% water). The number of patients discharged is also optimal. Then the LBO algorithm is applied to a more complex scenario of three venues and three hospitals. The optimization is performed in two stages: first determine the distributions for the ambulances and second determine the electricity and water distributions. The LBO algorithm is evaluated on three resource levels: 1) 50% electricity, 50% water and 50% ambulances, 2) 25% electricity, 50% water and 75% ambulances, 3) 50% electricity 25% water and 75% ambulances. Again, the LBO algorithm was able to accurately calculate the distribution ratios that enable the maximum number of transported patients and discharged patients. Finally the RL method is used as a comparison for the three hospitals with no transportation modeling scenario. The RL method computed through numerous iterations was finally able to capture the accurate choices and discharge the maximum number of patients. The RL method is very good, but it requires longer computation time. Overall, both methods have advantages and disadvantages that complement each other. For example, the Lagrange method could be used to calculate some distribution choices for the RL agent method. Then one can run the RL agent to find out the optimal results.

As mentioned previously, the I2Sim Decision Layer is still in conceptualization. The work presented in this thesis may eventually be part of the Decision Layer and used to calculate the optimum distribution ratios. Therefore this work can be used for decision support.

The results show that future works could be performed in the following areas:

- One key assumption throughout the simulations is the patients are all the same. That is they are at the same level of injury and require the same medical resources. There is no granularity on the patients (level of injury, age, gender, etc.) The hospitals are also assumed to have the same treatment facilities. After a disaster occurs, there would be a wide variety of patients: children and seniors, men and women, lightly injured and seriously injured. One way to classify the different levels of injuries is to model the tokens in I2Sim with an HRT. For example, level 1 is a healthy person, whereas level 5 is a deceased person. The hospitals can also have different functions, such as a general hospital, women's hospital and children's hospital. It would be interesting to extend this aspect of the research in future works to find out how granularity affects optimization decisions.
- Another aspect of future work is adding more resources. Some of the resources to be added include food, medicine, money, etc. Also we could add communication capabilities to each of the cells, and see how a cell would make local decisions if its communications were broken from the outside world.
- The LBO algorithm could be programmed into a Level 2 Matlab block and placed inside the I2Sim model. The inputs could be the available water and electricity, and the HRTs of the production cells. The outputs are the distribution ratios for water and electricity.
- This research work could also be extended to a general City Model to be built with I2Sim. The City Model is still in the process of conceptualization. But it would mimic that of a real city, such as Vancouver. The infrastructures include hospitals, residence buildings, office buildings, sports venues, roads, electrical substations and water stations, etc. It

would be really amazing to see how the optimization algorithm could be applied to the operation of a city both in peace time and disaster time.

Finally, as my supervisor Dr. K.D. Srivastava once said, "There is no end to research." The work presented in this thesis is neither the beginning nor the end, but a segment in the continuum of I2Sim research. I sincerely hope this thesis can serve as an inspiration for future great works to come.

Bibliography

[1] K. Wang, M. Bai, K.D. Srivastava and J. Marti. "Optimal Decision Maker Algorithm for Disaster Response Management with I2Sim Applications," *ISCRAM Conference*, 2012, pp. 1-5.

[2] UBC Joint Infrastructures Interdependencies Research Program. "The government of Canada announces $2.98 million to fund research on critical infrastructure interdependencies," Internet: http://www.ece.ubc.ca/~jiirp/JIIRP_Open_Publications/jiirp_i2c_001.pdf, 2005 [Accessed: June 2012].

[3] H. Lee. "Infrastructure Interdependencies Simulation (I2Sim) System Model and Toolbox." M.A.Sc. Thesis, University of BC, 2010.

[4] L. Liu. "Prototyping and Cells Modeling of the Infrastructure Interdependencies Simulator I2Sim." M.A.Sc. Thesis, University of BC, 2007.

[5] W. Wang and J. Wang. "I2Sim Japan Sendai Scenario." UBC ECE Power Lab Internal Report, 2012.

[6] C. Lopez. "Multi-Energy Systems Simulator for Hourly Management and Optimization of GHG Emissions and Fuel Costs." M.A.Sc. Thesis, University of BC, 2011.

[7] R. Ren. "I2Sim Financial Model and its Application to UBC's Living Lab Projects." M.A.Sc. Thesis, University of BC, 2011.

[8] P. Pederson, D. Dudenhoeffer, S. Hartley and M. Permann. "Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research." Idaho National Laboratory, Aug 2006, pp. 1-20.

[9] S.M. Rinaldi, J.P. Peerenboom and T.K. Kelly. "Identifying, Understanding and Analyzing Critical Infrastructure Interdependencies." *IEEE Control Systems Magazine*, Vol. 21, pp. 11-25, 2001.

[10] R. Zimmerman. "Decision-making and the Vulnerability of Interdependent Critical Infrastructure." *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, 2004, pp. 4059-4063.

[11] M. Khouj, C. Lopez, S. Sarkaria and J. Marti. "Disaster Management in Real Time Simulation using Machine Learning," in *Proc. Canadian Conference on Electrical and Computer Engineering*, 2011, pp. 1507-1510.

[12] E. Bagheri and A. Ghorbani. "Conceptualizing Critical Infrastructures as Service Oriented Complex Interdependent Systems." *Proc. International Conference of Information Technology and Management*, 2007, pp. 1-10.

[13] N. Weston, M.G. Balchanos, M.R. Koepp and D.N. Mavris. "Strategies for integrating models of interdependent subsystems of complex system-of-systems products." *Proc. of the 38th Southeastern Symposium on System Theory*, 2006, pp. 181-185.

[14] T.K. Kelly. "Infrastructure Interdependencies," PowerPoint Presentation, Internet: http://wpweb2.tepper.cmu.edu/ceic/presentations/Kelly.pdf [Accessed: May 2012].

[15] H.J. Garcia. "Multi-Hazard Risk Assessment: An Interdependency Approach," PhD Thesis, University of BC, 2010.

[16] O. Gursesli and A. Desrochers. "Modeling Infrastructure Interdependencies Using Petri Nets." *Proc. IEEE International Conference on System, Man and Cybernetics*, 2003, pp. 1506-1512.

[17] J.R. Marti, J. A. Hollman, C. Ventura and J. Jatskevich. "Dynamic Recovery of Critical Infrastructures: Real-time Temporal Coordination." *The International Journal Critical Infrastructures*, Vol. 4, pp. 18-30, 2008.

[18] J.R. Marti, C. Ventura, J.A. Hollman, K.D. Srivastava and H. Juarez. "I2Sim Modeling and Simulation for Scenario Development, Training and Real-Time Decision Support of Multiple Interdependent Critical Infrastructures During Large Emergencies." *RTA/MSG Conference on "How is Modelling and Simulation Meeting the Defense Challenges up to 2015?"*, 2008, pp. 1-13.

[19] University of BC. "I2Sim Technical Description and User Manual," Technical Manual, 2009, pp. 1-37.

[20] A. Singh. "Ontology for Decision Making in Emergency Response – Conceptual Design of Decision Layer in I2Sim," Internal PowerPoint Presentation, 2012.

[21] D. Li, G. Liu and Y. Gao. "Uncertainty Optimization Model for Emergency Resource Scheduling." *Proc. Second International Symposium on Knowledge Acquisition and Modeling*, 2009, pp. 55 – 58.

[22] G. Zhou and L. She. "Research on Scheduling Models of Emergency Resource." *Proc. Fourth International Conference on Intelligent Computation Technology and Automation*, 2011, pp. 1110-1113.

[23] A. Donner, T. Greiner-Mai and C. Adler. "Challenge Patient Dispatching in Mass Casualty Incidents," *Proc. 9th International ISCRAM Conference*, 2012, pp. 1-5.

[24] M. Khouj. "Decision Agent in Real Time Simulation using Reinforcement Learning and Neural Network," PhD Proposal, University of BC, 2009.

[25] D.G. Myers. "Psychology 8th Edition," Worth Publishers, New York, 2007, pp. 329-330.

[26] G. McDonell. "The Fred Kaiser Building." SABmag, pp. 16-20, Nov/Dec 2006.

[27] G. McDonell. "Thermal Mass System: A First in North America." Modern Hydronics Magazine, pp. 50 – 56, May/June 2006.

[28] Omicron. "The Fred Kaiser Building at UBC." Internet: http://www.omicronaec.com/docs/Project%20Profile_UBC%20Fred%20Kaiser%20Building%20EXPANDED%20SUSTAINABLE%20VERSION.pdf, [Accessed Jan. 9, 2012].

[29] Carmanah Corporation. "UBC Fred Kaiser Building." Internet: http://www.carmanah.com/grid-tie/installation-portfolio/ubc-fred-kaiser-building, [Accessed: May 9, 2012].

[30] Z. Tian and J. Love. "Radiant Slab Cooling: a Case Study of Building Energy Performance." *Proc. SimBuild Conference*, 2006, pp. 238-244.

[31] ESP-r. "ESP-r." Internet: http://www.esru.strath.ac.uk/Programs/ESP-r.htm. [Accessed Feb. 2012].

[32] EDSL Home. "Tas." Internet: http://www.edsl.net/main/. [Accessed Feb 2012].

[33] Trnsys. "Welcome to Trnsys." Internet: http://www.trnsys.com/. [Accessed Feb 2012]

[34] IDA. "IDA Indoor Climate and Energy 4.0." Internet: http://www.equa.se/eng.ice.html. [Accessed Feb 2012].

[35] US Dept of Energy. "EnergyPlus Energy Simulation Program." Internet: http://apps1.eere.energy.gov/buildings/energyplus/. [Accessed Feb 2012].

[36] BC Government. "BC Energy Plan." Internet: http://www.energyplan.gov.bc.ca/. [Accessed: May 2012].

[37] J. Zhu. "Introduction," in *Optimization of Power System Operations*, 1st ed. M.E. El-Hawary, Ed. Piscataway, NJ: IEEE Press, 2009, pp. 1-7.

[38] A.K. Dixit. "Optimization in Economic Theory," Oxford University Press, 1990, pp. 10-23.

[39] R. Baldick. "Algorithms for Linear Equality-Constrained Minimization," in *Applied Optimization Formulation and Algorithms for Engineering Systems*, 1st ed. Cambridge: Cambridge University Press, 2006, pp. 463-528.

[40] N.S. Rau. "Constrained Nonlinear Optimization," in Optimization Principles Practical Applications to the Operation and Markets of the Electric Power Industry, 1st. ed. P.M. Anderson, Ed. Piscataway, NJ: IEEE Press, 2003, pp. 177-244.

[41] M.D. Intriligator. "Optimization and Economic Theory," Siam, Philadelphia, 2002, pp. 28-38.

[42] H. Everett. "Generalized Lagrange Multiplier Method For Solving Problems of Optimum Allocation of Resources," Institute for Defense Analyses, 1962, pp. 399-417.

[43] J.A. Momoh. "Electric Power System Models," in *Electric Power System Applications of Optimization*, 1st. ed. H.L. Willis, Ed. New York: Marcel Dekker Inc., 2001, pp. 19-64.

[44] M.E. El-Hawary. "The Energy Control Center," in *Introduction to Electrical Power Systems*, 1st ed. M.E. El-Hawary, Ed. Piscataway, NJ: IEEE Press, 2008, pp. 305-366.

[45] K. Wang. "Decision Maker Algorithm," Internal PowerPoint Presentation, 2012.

[46] Electric Power & Energy System Group at UBC. "Disaster Response Decision Making Infrastructure," PowerPoint Presentation, 2012, pp. 1-81.

[47] M. Bai. "Modern Green Internship Report." Internal Report, 2011.

[48] J. Vandermaar. Personal Interview, 2012.

Appendices

Appendix A: Matlab Code for Three Hospitals No Transportation LBO Method

```
% 3 Hospital No Transportation Calculations
% Programmed by: Ming Bai

clc
clear all;

% Read in avail power and avail water
% Read in HRTs of hospitals
% Calculate elec solution
% Calculate water solution
% Resolve conflict
% Calculate distribution of elec and water
% Write the distributions back to the simulink scenario

a =
find_system('ming_3hosp_nodelay','FollowLinks','on','LookUnderMasks','none');
 % Read all blocks in model


% Calculate power and water distribution
x =  strmatch('ming_3hosp_nodelay',a);
% get the available power and water
avail_elec =  str2num(get_param('ming_3hosp_nodelay/BC Hydro','Output'));
avail_water = str2num(get_param('ming_3hosp_nodelay/Water
Supplier','Output'));

tot_elec = avail_elec; % total elec available before distributing to water
station
avail_elec = avail_elec - 0.01; % first need to take off the amount required
by water station
tot_water = avail_water;

name_hosp = ['ming_3hosp_nodelay/Hospital 1'; 'ming_3hosp_nodelay/Hospital 2';
'ming_3hosp_nodelay/Hospital 3']

% Read the HRT of the pump station, assume the number of rows are the same
% as the hospitals
q = get_param('ming_3hosp_nodelay/Water Pump','userdata');
Hrt_water = q.PhysicalMode.Table;
num_rows = size (Hrt_water, 1);

x2 =  strmatch('ming_3hosp_nodelay/Hospital',a);
num_hosp = size (x2, 1);

% Read hospital HRTs
hosp_hrt = [];  % an array to put all the hospitals HRTs together
% [ppl_output elec water ppl_out elec water ...]
for i=1:num_hosp
    hosp_arr(i) = get_param(name_hosp(i,:),'userdata');
```

```
    hosp_hrt = [hosp_hrt hosp_arr(i).PhysicalMode.Table];
end

% elec solution
% calculate efficiencies of hosp elec
for i = 1:4
    eff_H1_e(i) = (hosp_hrt(i,1) - hosp_hrt(i+1,1))/(hosp_hrt(i,2) - 
hosp_hrt(i+1,2));
    eff_H2_e(i) = (hosp_hrt(i,4) - hosp_hrt(i+1,4))/(hosp_hrt(i,5) - 
hosp_hrt(i+1,5));
    eff_H3_e(i) = (hosp_hrt(i,7) - hosp_hrt(i+1,7))/(hosp_hrt(i,8) - 
hosp_hrt(i+1,8));
end


% [eff elec_needed hosp# hrt_row#] 4 columns
H1e = [eff_H1_e' hosp_hrt(1:4,2) 1*ones(1,4)' [1 2 3 4]'];
H2e = [eff_H2_e' hosp_hrt(1:4,5) 2*ones(1,4)' [1 2 3 4]'];
H3e = [eff_H3_e' hosp_hrt(1:4,8) 3*ones(1,4)' [1 2 3 4]'];

% sort in ascending order, according to col 1 efficiency
B = sortrows([H1e; H2e; H3e], 1);

% sort in descending order
B = flipdim(B,1);

hosp_elec = []; % initialize amount of elec to each hosp

% [hosp# elec_amt hrt_row_num]
for i=1:num_hosp
    hosp_elec(i,1) = i; % number the hospital
    hosp_elec(i,2) = 0; % initially each hosp gets 0 elec
    hosp_elec(i,3) = 5; % initially each hosp operates at row 5 of hrt, or 
zero
end


hosp_elec


% Assign amount of elec to each hosp
for i = 1:length(B)
    if(B(i,2) <= avail_elec + hosp_elec(B(i,3),2)) % if corr power less than 
avail power + assigned power to this hosp
        if(hosp_elec(B(i,3),2) == 0) % if no power has been assigned yet
            hosp_elec(B(i,3),2) = B(i,2); % assign this amt of powr
            hosp_elec(B(i,3),3) = B(i,4); % update row num
        else % if power has been assigned
            avail_elec = avail_elec + hosp_elec(B(i,3),2);
            hosp_elec(B(i,3),2) = B(i,2); % assign this amt of powr
            hosp_elec(B(i,3),3) = B(i,4); % update row num
        end
        avail_elec = avail_elec - B(i,2); % update num of avail ambulance 
left
    end
```

```
end


hosp_elec

% Calculate the percentage to be sent back to distributor
elec_dist = []; % ambulance distribution %

for i=1:num_hosp+1
    if i == num_hosp+1
        elec_dist(num_hosp+1) = 0.01/tot_elec; % elec sent to water
distributor
    else
        elec_dist(i) = hosp_elec(i,2)/tot_elec;
    end
end

elec_dist

% convert the datatype of the distribution_power into char
% Inside the distributor block, it can only read the parameter in char
fact_elec=textconvert(elec_dist);

% sent the distribution ratios back

%set_param('ming_3venue_3hosp/dist_e','factors',fact_elec);




% water distribution
% calculate efficiencies of hosp water
for i = 1:4
    eff_H1_w(i) = (hosp_hrt(i,1) - hosp_hrt(i+1,1))/(hosp_hrt(i,3) -
hosp_hrt(i+1,3));
    eff_H2_w(i) = (hosp_hrt(i,4) - hosp_hrt(i+1,4))/(hosp_hrt(i,6) -
hosp_hrt(i+1,6));
    eff_H3_w(i) = (hosp_hrt(i,7) - hosp_hrt(i+1,7))/(hosp_hrt(i,9) -
hosp_hrt(i+1,9));
end

% [eff water_needed hosp# hrt_row#] 4 columns
H1w = [eff_H1_w' hosp_hrt(1:4,3) 1*ones(1,4)' [1 2 3 4]'];
H2w = [eff_H2_w' hosp_hrt(1:4,6) 2*ones(1,4)' [1 2 3 4]'];
H3w = [eff_H3_w' hosp_hrt(1:4,9) 3*ones(1,4)' [1 2 3 4]'];


% sort in ascending order, according to col 1 efficiency
C = sortrows([H1w; H2w; H3w], 1);

% sort in descending order
C = flipdim(C,1);
```

```
hosp_water = []; % initialize amount of elec to each hosp

% [hosp# elec_amt hrt_row_num]
for i=1:num_hosp
    hosp_water(i,1) = i; % number the hospital
    hosp_water(i,2) = 0; % initially each hosp gets 0 elec
    hosp_water(i,3) = 5; % initially each hosp operates at row 5 of hrt, or
zero
end


hosp_water

% Assign amount of water to each hosp
for i = 1:length(C)
    if(C(i,2) <= avail_water + hosp_water(C(i,3),2)) % if corr power less
than avail power + assigned power to this hosp
        if(hosp_water(C(i,3),2) == 0) % if no power has been assigned yet
            hosp_water(C(i,3),2) = C(i,2); % assign this amt of powr
            hosp_water(C(i,3),3) = C(i,4); % update row num
        else % if power has been assigned
            avail_water = avail_water + hosp_water(C(i,3),2);
            hosp_water(C(i,3),2) = C(i,2); % assign this amt of powr
            hosp_water(C(i,3),3) = C(i,4); % update row num
        end
        avail_water = avail_water - C(i,2); % update num of avail ambulance
left
    end
end


hosp_water

% Calculate the percentage to be sent back to distributor
water_dist = []; % ambulance distribution %

for i=1:num_hosp
    water_dist(i) = hosp_water(i,2)/tot_water; % elec sent to water
distributor
end

water_dist

% convert the datatype of the distribution_power into char
% Inside the distributor block, it can only read the parameter in char
fact_water=textconvert(water_dist);

% sent the distribution ratios back
%set_param('ming_3venue_3hosp/dist_w','factors',fact_water);


% check for conflicts

% extract row nums
```

```
row_e = hosp_elec(:,3);
row_w = hosp_water(:,3);

conf = 0;

if(row_e(1) == row_w(1) && row_e(2) && row_w(2) && row_e(3) && row_w(3)) % no
conflict
    conf = 1;
elseif (row_e(1) > row_w(1) && row_e(2) > row_w(2) && row_e(3) > row_w(3)) %
elec is limiter
    conf = 2;
elseif (row_w(1) > row_e(1) && row_w(2) > row_e(2) && row_w(3) > row_e(3)) %
water is limiter
    conf = 3;
else                                                                          %
need to list all poss
    conf = 4;
end

conf

switch conf
    case 1 % no conflict
        set_param('ming_3hosp_nodelay/dist_w','factors',fact_water);
        set_param('ming_3hosp_nodelay/dist_e','factors',fact_elec);

    case 2 % elec is limiter
        set_param('ming_3hosp_nodelay/dist_e','factors',fact_elec);

        for i = 1:num_hosp
            row_w(i) = row_e(i);
            hosp_water(i, 3) = row_w(i);     % set water rows to elec rows
            hosp_water(i, 2) = hosp_hrt(row_w(i), 3*i); % set water amount
            water_dist(i) = hosp_water(i,2)/tot_water; % calculate ratio
        end
        fact_water=textconvert(water_dist);
        set_param('ming_3hosp_nodelay/dist_w','factors',fact_water);

    case 3 % water is limiter
        set_param('ming_3hosp_nodelay/dist_w','factors',fact_water);

        for i = 1:num_hosp
            row_e(i) = row_w(i);
            hosp_elec(i,3) = row_e(i);
            hosp_elec(i,2) = hosp_hrt(row_e(i), 2*i);
            elec_dist(i) = hosp_elec(i,2)/tot_elec;
        end
        hosp_elec(num_hosp + 1) = 0.01/tot_elec;
        fact_elec=textconvert(elec_dist);
        set_param('ming_3hosp_nodelay/dist_e','factors',fact_elec);

    case 4 % need to list all poss
        % generate all row poss
        row_poss = [hosp_water(1,3) hosp_water(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_elec(2,3) hosp_elec(3,3)
```

```
                    hosp_water(1,3) hosp_water(2,3) hosp_elec(3,3)
                    hosp_water(1,3) hosp_elec(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_water(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_elec(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_water(1,3) hosp_elec(3,3)
                    hosp_water(1,3) hosp_elec(2,3) hosp_elec(3,3)];


        %poss1 = [[1 2 3]' row_poss(:,1) [hosp_hrt(row_poss(1,1),2)
hosp_hrt(row_poss(2,1),5) hosp_hrt(row_poss(3,1),8)]'
[hosp_hrt(row_poss(1,1),3) hosp_hrt(row_poss(2,1),6)
hosp_hrt(row_poss(3,1),9)]  ];
        % [hosp# row# ppl elec water] 5 columns
        for i=1:3
            poss1(i,1) =i;
            poss1(i,2) =row_poss(1,i);
            poss1(i,3:5) = hosp_hrt(row_poss(1,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss2(i,1) =i;
            poss2(i,2) =row_poss(2,i);
            poss2(i,3:5) = hosp_hrt(row_poss(2,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss3(i,1) =i;
            poss3(i,2) =row_poss(3,i);
            poss3(i,3:5) = hosp_hrt(row_poss(3,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss4(i,1) =i;
            poss4(i,2) =row_poss(4,i);
            poss4(i,3:5) = hosp_hrt(row_poss(4,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss5(i,1) =i;
            poss5(i,2) =row_poss(5,i);
            poss5(i,3:5) = hosp_hrt(row_poss(5,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss6(i,1) =i;
            poss6(i,2) =row_poss(6,i);
            poss6(i,3:5) = hosp_hrt(row_poss(6,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss7(i,1) =i;
            poss7(i,2) =row_poss(7,i);
            poss7(i,3:5) = hosp_hrt(row_poss(7,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss8(i,1) =i;
            poss8(i,2) =row_poss(8,i);
            poss8(i,3:5) = hosp_hrt(row_poss(8,i),(1+3*(i-1)):(1+3*(i-1)+2));
        end
        tot_poss = [poss1 poss2 poss3 poss4 poss5 poss6 poss7 poss8]; % put
all poss into 1 matrix

        % now check for the optimal possibility
        opt = [];
        max_ppl = 0;
        for i=1:8
            % check if violate water and/or elec constr
            % if not, see what is the ppl output value
```

```
            elec = 0;
            wat = 0;
            ppl = 0;
            for j = 1:3
                elec = elec + tot_poss(j, 4+5*(i-1));
                wat = wat + tot_poss(j, 5+5*(i-1));
                ppl = ppl + tot_poss(j, 3+5*(i-1));
            end

            %max_ppl = 0;
            if(tot_water >= wat && tot_elec >= elec) % if no resource
constraint violated
                if(ppl >= max_ppl) % if this choice is bigger than the
existing max ppl
                    opt = tot_poss(1:3,(1+5*(i-1)):((1+5*(i-1)+4)));
                    max_ppl = ppl
                end
            end

        end
        opt
        disp('output people rate is')
        p = 0;
        for i = 1:3
            p = p + opt(i,3);
        end
        p
        % after all that, we get the optimal allocation, calculate water
        % and elec distr ratios
        % water distr ratio
        for i = 1:num_hosp
            water_dist(i) = opt(i,5)/tot_water; % water distribution calc
        end
        fact_water=textconvert(water_dist);
        set_param('ming_3hosp_nodelay/dist_w','factors',fact_water);

        % elec distr ratio
        for i = 1:4
            if( i == 4)
                elec_dist(i) = 0.01/tot_elec;
            else
                elec_dist(i) = opt(i,4)/tot_elec;
            end
        end
        fact_elec=textconvert(elec_dist);
        set_param('ming_3hosp_nodelay/dist_e','factors',fact_elec);

end
```

Appendix B: Matlab Code for Three Venue Three Hospital with Transportation LBO Method

```
% Programmed by: Ming Bai
% 3 Venue 3 Hospital Calculation

clc
clear all;

% First calculate distribution of ambulances
a =
find_system('ming_3venue_3hosp','FollowLinks','on','LookUnderMasks','none');  %
Read all blocks in model
x =  strmatch('ming_3venue_3hosp/Control System',a);

num_chans = 9; % number of channels
avail_amb = str2num(get_param('ming_3venue_3hosp/Ambulance','Output'))
tot_amb = avail_amb; % set total amb

name_chan = ['ming_3venue_3hosp/Control System 1'; 'ming_3venue_3hosp/Control
System 2'; 'ming_3venue_3hosp/Control System 3'; 'ming_3venue_3hosp/Control
System 4'; 'ming_3venue_3hosp/Control System 5';
             'ming_3venue_3hosp/Control System 6'; 'ming_3venue_3hosp/Control
System 7'; 'ming_3venue_3hosp/Control System 8'; 'ming_3venue_3hosp/Control
System 9' ];

Channels = [];
for i = 1:num_chans
    Chan_arr (i) = get_param(name_chan(i,:),'userdata');

    Channels = [Channels Chan_arr(i).PhysicalMode.Table];
end;

% calculate the efficiencies of channels
for i = 1:4
        eff_C1(i) = (Channels(i,1) - Channels(i+1,1))/(Channels(i,2) -
Channels(i+1,2));
        eff_C2(i) = (Channels(i,3) - Channels(i+1,3))/(Channels(i,4) -
Channels(i+1,4));
        eff_C3(i) = (Channels(i,5) - Channels(i+1,5))/(Channels(i,6) -
Channels(i+1,6));
        eff_C4(i) = (Channels(i,7) - Channels(i+1,7))/(Channels(i,8) -
Channels(i+1,8));
        eff_C5(i) = (Channels(i,9) - Channels(i+1,9))/(Channels(i,10) -
Channels(i+1,10));
        eff_C6(i) = (Channels(i,11) - Channels(i+1,11))/(Channels(i,12) -
Channels(i+1,12));
        eff_C7(i) = (Channels(i,13) - Channels(i+1,13))/(Channels(i,14) -
Channels(i+1,14));
        eff_C8(i) = (Channels(i,15) - Channels(i+1,15))/(Channels(i,16) -
Channels(i+1,16));
        eff_C9(i) = (Channels(i,17) - Channels(i+1,17))/(Channels(i,18) -
Channels(i+1,18));
end
```

```
% [efficiency ambulance_needed channel#]
C1 = [eff_C1' Channels(1:4,2) 1*ones(1,4)'];
C2 = [eff_C2' Channels(1:4,4) 2*ones(1,4)'];
C3 = [eff_C3' Channels(1:4,6) 3*ones(1,4)'];
C4 = [eff_C4' Channels(1:4,8) 4*ones(1,4)'];
C5 = [eff_C5' Channels(1:4,10) 5*ones(1,4)'];
C6 = [eff_C6' Channels(1:4,12) 6*ones(1,4)'];
C7 = [eff_C7' Channels(1:4,14) 7*ones(1,4)'];
C8 = [eff_C8' Channels(1:4,16) 8*ones(1,4)'];
C9 = [eff_C9' Channels(1:4,18) 9*ones(1,4)'];

% sort in ascending order, according to col 1 efficiency
A = sortrows([C1; C2; C3; C4; C5; C6; C7; C8; C9], 1);

% sort in descending order
A = flipdim(A,1);

chan_amb = []; % channel ambulance dispatch

for i = 1:num_chans
    chan_amb(i,1) = i; % number the channel
    chan_amb(i,2) = 0; % initially each channel assigned no ambulance
end

% Assign number of ambulances
for i = 1:length(A)
    if(A(i,2) <= avail_amb + chan_amb(A(i,3),2)) % if corr ambulance less
than avail amb + assigned amb to this channel
        if(chan_amb(A(i,3),2) == 0) % if no amb has been assigned yet
            chan_amb(A(i,3),2) = A(i,2);
        else % if amb has been assigned
            avail_amb = avail_amb + chan_amb(A(i,3),2);
            chan_amb(A(i,3),2) = A(i,2);
        end
        avail_amb = avail_amb - A(i,2); % update num of avail ambulance left
    end
end

chan_amb

% Calculate the percentage to be sent back to distributor
amb_dist = []; % ambulance distribution %

for i=1:num_chans
    amb_dist(i) = chan_amb(i,2)/tot_amb;
end

amb_dist

% convert the datatype of the distribution_power into char
% Inside the distributor block, it can only read the parameter in char
fact_amb=textconvert(amb_dist);
```

```
% sent the distribution ratios back
set_param ('ming_3venue_3hosp/dist_a','factors',fact_amb);

% Calculate power and water distribution
x =  strmatch('ming_3venue_3hosp/Hospital',a);
% get the available power and water
avail_elec =  str2num(get_param('ming_3venue_3hosp/BC Hydro','Output'));
avail_water = str2num(get_param('ming_3venue_3hosp/Water Supplier','Output'));

tot_elec = avail_elec; % total elec available before distributing to water
station
avail_elec = avail_elec - 0.01; % first need to take off the amount required
by water station
tot_water = avail_water;

name_hosp = ['ming_3venue_3hosp/Hospital 1'; 'ming_3venue_3hosp/Hospital 2';
'ming_3venue_3hosp/Hospital 3']

% Read the HRT of the pump station, assume the number of rows are the same
% as the hospitals
q = get_param('ming_3venue_3hosp/Water Pump','userdata');
Hrt_water = q.PhysicalMode.Table;
num_rows = size (Hrt_water, 1);

x2 =  strmatch('ming_3venue_3hosp/Hospital',a);
num_hosp = size (x2, 1);

% Read hospital HRTs
hosp_hrt = [];  % an array to put all the hospitals HRTs together
% [ppl_output elec water ppl_out elec water ...]
for i=1:num_hosp
    hosp_arr(i) = get_param(name_hosp(i,:),'userdata');
    hosp_hrt = [hosp_hrt hosp_arr(i).PhysicalMode.Table];
end

% elec solution
% calculate efficiencies of hosp elec
for i = 1:4
    eff_H1_e(i) = (hosp_hrt(i,1) - hosp_hrt(i+1,1))/(hosp_hrt(i,2) -
hosp_hrt(i+1,2));
    eff_H2_e(i) = (hosp_hrt(i,4) - hosp_hrt(i+1,4))/(hosp_hrt(i,5) -
hosp_hrt(i+1,5));
    eff_H3_e(i) = (hosp_hrt(i,7) - hosp_hrt(i+1,7))/(hosp_hrt(i,8) -
hosp_hrt(i+1,8));
end


% [eff elec_needed hosp# hrt_row#] 4 columns
H1e = [eff_H1_e' hosp_hrt(1:4,2) 1*ones(1,4)' [1 2 3 4]'];
H2e = [eff_H2_e' hosp_hrt(1:4,5) 2*ones(1,4)' [1 2 3 4]'];
H3e = [eff_H3_e' hosp_hrt(1:4,8) 3*ones(1,4)' [1 2 3 4]'];

% sort in ascending order, according to col 1 efficiency
B = sortrows([H1e; H2e; H3e], 1);
```

```
% sort in descending order
B = flipdim(B,1);

hosp_elec = []; % initialize amount of elec to each hosp

% [hosp# elec_amt hrt_row_num]
for i=1:num_hosp
    hosp_elec(i,1) = i; % number the hospital
    hosp_elec(i,2) = 0; % initially each hosp gets 0 elec
    hosp_elec(i,3) = 5; % initially each hosp operates at row 5 of hrt, or
zero
end

% Assign amount of elec to each hosp
for i = 1:length(B)
    if(B(i,2) <= avail_elec + hosp_elec(B(i,3),2)) % if corr power less than
avail power + assigned power to this hosp
        if(hosp_elec(B(i,3),2) == 0) % if no power has been assigned yet
            hosp_elec(B(i,3),2) = B(i,2); % assign this amt of powr
            hosp_elec(B(i,3),3) = B(i,4); % update row num
        else % if power has been assigned
            avail_elec = avail_elec + hosp_elec(B(i,3),2);
            hosp_elec(B(i,3),2) = B(i,2); % assign this amt of powr
            hosp_elec(B(i,3),3) = B(i,4); % update row num
        end
        avail_elec = avail_elec - B(i,2); % update num of avail ambulance
left
    end
end


hosp_elec

% Calculate the percentage to be sent back to distributor
elec_dist = []; % ambulance distribution %

for i=1:num_hosp+1
    if i == num_hosp+1
        elec_dist(num_hosp+1) = 0.01/tot_elec; % elec sent to water
distributor
    else
        elec_dist(i) = hosp_elec(i,2)/tot_elec;
    end
end

elec_dist

% convert the datatype of the distribution_power into char
% Inside the distributor block, it can only read the parameter in char
fact_elec=textconvert(elec_dist);

% sent the distribution ratios back

% water distribution
% calculate efficiencies of hosp water
```

```
for i = 1:4
    eff_H1_w(i) = (hosp_hrt(i,1) - hosp_hrt(i+1,1))/(hosp_hrt(i,3) -
hosp_hrt(i+1,3));
    eff_H2_w(i) = (hosp_hrt(i,4) - hosp_hrt(i+1,4))/(hosp_hrt(i,6) -
hosp_hrt(i+1,6));
    eff_H3_w(i) = (hosp_hrt(i,7) - hosp_hrt(i+1,7))/(hosp_hrt(i,9) -
hosp_hrt(i+1,9));
end

% [eff water_needed hosp# hrt_row#] 4 columns
H1w = [eff_H1_w' hosp_hrt(1:4,3) 1*ones(1,4)' [1 2 3 4]'];
H2w = [eff_H2_w' hosp_hrt(1:4,6) 2*ones(1,4)' [1 2 3 4]'];
H3w = [eff_H3_w' hosp_hrt(1:4,9) 3*ones(1,4)' [1 2 3 4]'];


% sort in ascending order, according to col 1 efficiency
C = sortrows([H1w; H2w; H3w], 1);

% sort in descending order
C = flipdim(C,1);

hosp_water = []; % initialize amount of elec to each hosp

% [hosp# elec_amt hrt_row_num]
for i=1:num_hosp
    hosp_water(i,1) = i; % number the hospital
    hosp_water(i,2) = 0; % initially each hosp gets 0 elec
    hosp_water(i,3) = 5; % initially each hosp operates at row 5 of hrt, or
zero
end


% Assign amount of water to each hosp
for i = 1:length(C)
    if(C(i,2) <= avail_water + hosp_water(C(i,3),2)) % if corr power less
than avail power + assigned power to this hosp
        if(hosp_water(C(i,3),2) == 0) % if no power has been assigned yet
            hosp_water(C(i,3),2) = C(i,2); % assign this amt of powr
            hosp_water(C(i,3),3) = C(i,4); % update row num
        else % if power has been assigned
            avail_water = avail_water + hosp_water(C(i,3),2);
            hosp_water(C(i,3),2) = C(i,2); % assign this amt of powr
            hosp_water(C(i,3),3) = C(i,4); % update row num
        end
        avail_water = avail_water - C(i,2); % update num of avail ambulance
left
    end
end


hosp_water

% Calculate the percentage to be sent back to distributor
water_dist = []; % ambulance distribution %
```

```
for i=1:num_hosp
    water_dist(i) = hosp_water(i,2)/tot_water; % elec sent to water
distributor
end

water_dist

% convert the datatype of the distribution_power into char
% Inside the distributor block, it can only read the parameter in char
fact_water=textconvert(water_dist);

% sent the distribution ratios back

% check for conflicts

% extract row nums
row_e = hosp_elec(:,3);
row_w = hosp_water(:,3);

conf = 0;

if(row_e(1) == row_w(1) && row_e(2) && row_w(2) && row_e(3) && row_w(3)) % no
conflict
    conf = 1;
elseif (row_e(1) > row_w(1) && row_e(2) > row_w(2) && row_e(3) > row_w(3)) %
elec is limiter
    conf = 2;
elseif (row_w(1) > row_e(1) && row_w(2) > row_e(2) && row_w(3) > row_e(3)) %
water is limiter
    conf = 3;
else                                                                          %
need to list all poss
    conf = 4;
end

conf

switch conf
    case 1 % no conflict
        set_param('ming_3venue_3hosp/dist_w','factors',fact_water);
        set_param('ming_3venue_3hosp/dist_e','factors',fact_elec);

    case 2 % elec is limiter
        set_param('ming_3venue_3hosp/dist_e','factors',fact_elec);

        for i = 1:num_hosp
            row_w(i) = row_e(i);
            hosp_water(i, 3) = row_w(i);     % set water rows to elec rows
            hosp_water(i, 2) = hosp_hrt(row_w(i), 3*i); % set water amount
            water_dist(i) = hosp_water(i,2)/tot_water; % calculate ratio
        end
        fact_water=textconvert(water_dist);
        set_param('ming_3venue_3hosp/dist_w','factors',fact_water);
```

```
    case 3 % water is limiter
        set_param('ming_3venue_3hosp/dist_w','factors',fact_water);

        for i = 1:num_hosp
            row_e(i) = row_w(i);
            hosp_elec(i,3) = row_e(i);
            hosp_elec(i,2) = hosp_hrt(row_e(i), 2*i);
            elec_dist(i) = hosp_elec(i,2)/tot_elec;
        end
        hosp_elec(num_hosp + 1) = 0.01/tot_elec;
        fact_elec=textconvert(elec_dist);
        set_param('ming_3venue_3hosp/dist_e','factors',fact_elec);

    case 4 % need to list all poss
        % generate all row poss
        row_poss = [hosp_water(1,3) hosp_water(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_elec(2,3) hosp_elec(3,3)
                    hosp_water(1,3) hosp_water(2,3) hosp_elec(3,3)
                    hosp_water(1,3) hosp_elec(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_water(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_elec(2,3) hosp_water(3,3)
                    hosp_elec(1,3) hosp_water(1,3) hosp_elec(3,3)
                    hosp_water(1,3) hosp_elec(2,3) hosp_elec(3,3)];


        %poss1 = [[1 2 3]' row_poss(:,1) [hosp_hrt(row_poss(1,1),2)
hosp_hrt(row_poss(2,1),5) hosp_hrt(row_poss(3,1),8)]'
[hosp_hrt(row_poss(1,1),3) hosp_hrt(row_poss(2,1),6)
hosp_hrt(row_poss(3,1),9)]  ];
        % [hosp# row# ppl elec water] 5 columns
        for i=1:3
            poss1(i,1) =i;
            poss1(i,2) =row_poss(1,i);
            poss1(i,3:5) = hosp_hrt(row_poss(1,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss2(i,1) =i;
            poss2(i,2) =row_poss(2,i);
            poss2(i,3:5) = hosp_hrt(row_poss(2,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss3(i,1) =i;
            poss3(i,2) =row_poss(3,i);
            poss3(i,3:5) = hosp_hrt(row_poss(3,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss4(i,1) =i;
            poss4(i,2) =row_poss(4,i);
            poss4(i,3:5) = hosp_hrt(row_poss(4,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss5(i,1) =i;
            poss5(i,2) =row_poss(5,i);
            poss5(i,3:5) = hosp_hrt(row_poss(5,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss6(i,1) =i;
            poss6(i,2) =row_poss(6,i);
            poss6(i,3:5) = hosp_hrt(row_poss(6,i),(1+3*(i-1)):(1+3*(i-1)+2));
```

```
            poss7(i,1) =i;
            poss7(i,2) =row_poss(7,i);
            poss7(i,3:5) = hosp_hrt(row_poss(7,i),(1+3*(i-1)):(1+3*(i-1)+2));

            poss8(i,1) =i;
            poss8(i,2) =row_poss(8,i);
            poss8(i,3:5) = hosp_hrt(row_poss(8,i),(1+3*(i-1)):(1+3*(i-1)+2));
        end
        tot_poss = [poss1 poss2 poss3 poss4 poss5 poss6 poss7 poss8]; % put
all poss into 1 matrix

        % now check for the optimal possibility
        opt = [];
        max_ppl = 0;
        for i=1:8
            % check if violate water and/or elec constr
            % if not, see what is the ppl output value
            elec = 0;
            wat = 0;
            ppl = 0;
            for j = 1:3
                elec = elec + tot_poss(j, 4+5*(i-1));
                wat = wat + tot_poss(j, 5+5*(i-1));
                ppl = ppl + tot_poss(j, 3+5*(i-1));
            end

            %max_ppl = 0;
            if(tot_water >= wat && tot_elec >= elec) % if no resource
constraint violated
                if(ppl >= max_ppl) % if this choice is bigger than the
existing max ppl
                    opt = tot_poss(1:3,(1+5*(i-1)):((1+5*(i-1)+4)));
                    max_ppl = ppl
                end
            end

        end
        opt
        disp('output people rate is')
        p = 0;
        for i = 1:3
            p = p + opt(i,3);
        end
        p
        % after all that, we get the optimal allocation, calculate water
        % and elec distr ratios
        % water distr ratio
        for i = 1:num_hosp
            water_dist(i) = opt(i,5)/tot_water; % water distribution calc
        end
        fact_water=textconvert(water_dist);
        set_param('ming_3venue_3hosp/dist_w','factors',fact_water);

        % elec distr ratio
        for i = 1:4
```

```
            if( i == 4)
                elec_dist(i) = 0.01/tot_elec;
            else
                elec_dist(i) = opt(i,4)/tot_elec;
            end
        end
        fact_elec=textconvert(elec_dist);
        set_param('ming_3venue_3hosp/dist_e','factors',fact_elec);

end
```

Appendix C: Matlab Code for Three Hospitals No Transportation RL Method [24]

```
function agent_level_2m(block)
% Level-2 M file S-Function for times two demo.
%   Copyright 1990-2004 The MathWorks, Inc.
%   $Revision: 1.1.6.1 $
%   Programmed by: Cesar Lopez clopez@ece.ubc.ca
%   Modified by: Ming Bai, with permission from Cesar Lopez
%   This test case connects DAARTS agent to 3 hospital with no transportation
%   scenario
  setup(block);

%endfunction
function setup(block)

  %% Register number of input and output ports
  block.NumInputPorts  = 12;
  block.NumOutputPorts = 7;

  %% Setup functional port properties to dynamically
  %% inherited.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

 % Allow multidimensional signals
  block.AllowSignalsWithMoreThan2D = true;

 % Register parameters
  block.NumDialogPrms = 3;
  block.DialogPrmsTunable = {'Tunable', 'Tunable', 'Tunable'};
  %Parameters are ={Nbr of states , Nbr of actions , Nbr of PMs}

 % Override input port properties
  for i=1:block.NumInputPorts
    block.InputPort(i).Dimensions  = 1;
    block.InputPort(i).DatatypeID  = 0;  % double
    block.InputPort(i).Complexity  = 'Real';
    block.InputPort(i).DirectFeedthrough = true;
  end

% Override output port properties
%   for i=1:block.NumOutputPorts-1
%     block.OutputPort(i).Dimensions       = 1;
%     block.OutputPort(i).DatatypeID  = 0; % double
%     block.OutputPort(i).Complexity  = 'Real';
%   end

  block.OutputPort(1).Dimensions  = 4;%combination for Distributor 1;
  block.OutputPort(1).DatatypeID  = 0; % double
  block.OutputPort(1).Complexity  = 'Real';

  block.OutputPort(2).Dimensions  = 3;%combination for Distributor 2;
  block.OutputPort(2).DatatypeID  = 0; % double
  block.OutputPort(2).Complexity  = 'Real';
```

```
  block.OutputPort(3).Dimensions  = 1;%rewards;
  block.OutputPort(3).DatatypeID  = 0; % double
  block.OutputPort(3).Complexity  = 'Real';

  block.OutputPort(4).Dimensions  =  [ block.DialogPrm(2).Data,
block.DialogPrm(1).Data ]; %LUT
  block.OutputPort(4).DatatypeID  = 0; % double
  block.OutputPort(4).Complexity  = 'Real';

  block.OutputPort(5).Dimensions  = 2; %PrvSt  Recall this structure is a
vector lenght 2 [state|action]
  block.OutputPort(5).DatatypeID  = 0; % double
  block.OutputPort(5).Complexity  = 'Real';

  block.OutputPort(6).Dimensions  = 2; %CurSt  Recall this structure is a
vector lenght 2 [state|action]
  block.OutputPort(6).DatatypeID  = 0; % double
  block.OutputPort(6).Complexity  = 'Real';

  block.OutputPort(7).Dimensions  = 1; %Exploratory signal
  block.OutputPort(7).DatatypeID  = 0; % double
  block.OutputPort(7).Complexity  = 'Real';

  %% Set block sample time to inherited
  block.SampleTimes = [-1 0];

  %% Run accelerator on TLC
  block.SetAccelRunOnTLC(true);

  %% Register methods
  block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
  block.RegBlockMethod('InitializeConditions',    @InitConditions);
  block.RegBlockMethod('SetInputPortSamplingMode',@SetInputPortSamplingMode);
  %block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
  block.RegBlockMethod('Terminate', @Terminate);
  block.RegBlockMethod('Outputs', @Output);
%endfunction

function DoPostPropSetup(block)

  %% Setup Dwork
  block.NumDworks = 21+block.DialogPrm(2).Data;

  block.Dwork(1).Name = 'NbrStates';%Number of states
  block.Dwork(1).Dimensions      = 1;
  block.Dwork(1).DatatypeID      = 0;
  block.Dwork(1).Complexity      = 'Real';
  block.Dwork(1).UsedAsDiscState = true;

  block.Dwork(2).Name = 'discharged_Hist';%previous number of discharged
patients
  block.Dwork(2).Dimensions      = 1;
  block.Dwork(2).DatatypeID      = 0;
```

```
  block.Dwork(2).Complexity      = 'Real';
  block.Dwork(2).UsedAsDiscState = true;

  block.Dwork(3).Name = 'PreviousSt';%Previous visited state to update value
Recall this structure is a vector lenght 2 [state|action]
  block.Dwork(3).Dimensions      = 2;
  block.Dwork(3).DatatypeID      = 0;
  block.Dwork(3).Complexity      = 'Real';
  block.Dwork(3).UsedAsDiscState = true;

  block.Dwork(4).Name = 'CurrentSt';%Current chosen state
Recall this structure is a vector length 2 [state|action]
  block.Dwork(4).Dimensions      = 2;
  block.Dwork(4).DatatypeID      = 0;
  block.Dwork(4).Complexity      = 'Real';
  block.Dwork(4).UsedAsDiscState = true;

  block.Dwork(5).Name = 'learning'; %Learning rate
  block.Dwork(5).Dimensions      = 1;
  block.Dwork(5).DatatypeID      = 0;
  block.Dwork(5).Complexity      = 'Real';
  block.Dwork(5).UsedAsDiscState = true;

  block.Dwork(6).Name = 'discount';%Discount factor
  block.Dwork(6).Dimensions      = 1;
  block.Dwork(6).DatatypeID      = 0;
  block.Dwork(6).Complexity      = 'Real';
  block.Dwork(6).UsedAsDiscState = true;

  block.Dwork(7).Name = 'Dis1Comb';%Number of combinations for distr 1
  block.Dwork(7).Dimensions      = 1;
  block.Dwork(7).DatatypeID      = 0;
  block.Dwork(7).Complexity      = 'Real';
  block.Dwork(7).UsedAsDiscState = true;

  block.Dwork(8).Name = 'Dis2Comb';%Number of combinations for distr 2
  block.Dwork(8).Dimensions      = 1;
  block.Dwork(8).DatatypeID      = 0;
  block.Dwork(8).Complexity      = 'Real';
  block.Dwork(8).UsedAsDiscState = true;

  block.Dwork(9).Name = 'Dis1StartPoint';%Index of first Dwork of
combinations for distr 1
  block.Dwork(9).Dimensions      = 1;
  block.Dwork(9).DatatypeID      = 0;
  block.Dwork(9).Complexity      = 'Real';
  block.Dwork(9).UsedAsDiscState = true;

  block.Dwork(10).Name = 'Dis2StartPoint';%Index of first Dwork of
combinations for distr 2
  block.Dwork(10).Dimensions      = 1;
  block.Dwork(10).DatatypeID      = 0;
  block.Dwork(10).Complexity      = 'Real';
  block.Dwork(10).UsedAsDiscState = true;
```

```
block.Dwork(11).Name = 'Distr1Comb1';%Combination 1 for Distributor 1
block.Dwork(11).Dimensions      = 4;
block.Dwork(11).DatatypeID      = 0;
block.Dwork(11).Complexity      = 'Real';
block.Dwork(11).UsedAsDiscState = true;

block.Dwork(12).Name = 'Distr1Comb2';%Combination 2 for Distributor 1
block.Dwork(12).Dimensions      = 4;
block.Dwork(12).DatatypeID      = 0;
block.Dwork(12).Complexity      = 'Real';
block.Dwork(12).UsedAsDiscState = true;

block.Dwork(13).Name = 'Distr1Comb3';%Combination 3 for Distributor 1
block.Dwork(13).Dimensions      = 4;
block.Dwork(13).DatatypeID      = 0;
block.Dwork(13).Complexity      = 'Real';
block.Dwork(13).UsedAsDiscState = true;

block.Dwork(14).Name = 'Distr1Comb4';%Combination 4 for Distributor 1
block.Dwork(14).Dimensions      = 4;
block.Dwork(14).DatatypeID      = 0;
block.Dwork(14).Complexity      = 'Real';
block.Dwork(14).UsedAsDiscState = true;

block.Dwork(15).Name = 'Distr1Comb5';%Combination 5 for Distributor 1
block.Dwork(15).Dimensions      = 4;
block.Dwork(15).DatatypeID      = 0;
block.Dwork(15).Complexity      = 'Real';
block.Dwork(15).UsedAsDiscState = true;

block.Dwork(16).Name = 'Distr2Comb1';%Combination 1 for Distributor 2
block.Dwork(16).Dimensions      = 3;
block.Dwork(16).DatatypeID      = 0;
block.Dwork(16).Complexity      = 'Real';
block.Dwork(16).UsedAsDiscState = true;

block.Dwork(17).Name = 'Distr2Comb2';%Combination 2 for Distributor 2
block.Dwork(17).Dimensions      = 3;
block.Dwork(17).DatatypeID      = 0;
block.Dwork(17).Complexity      = 'Real';
block.Dwork(17).UsedAsDiscState = true;

block.Dwork(18).Name = 'Distr2Comb3';%Combination 3 for Distributor 2
block.Dwork(18).Dimensions      = 3;
block.Dwork(18).DatatypeID      = 0;
block.Dwork(18).Complexity      = 'Real';
block.Dwork(18).UsedAsDiscState = true;

block.Dwork(19).Name = 'Distr2Comb4';%Combination 4 for Distributor 2
block.Dwork(19).Dimensions      = 3;
block.Dwork(19).DatatypeID      = 0;
block.Dwork(19).Complexity      = 'Real';
block.Dwork(19).UsedAsDiscState = true;

block.Dwork(20).Name = 'Distr2Comb5';%Combination 4 for Distributor 2
```

```
  block.Dwork(20).Dimensions      = 3;
  block.Dwork(20).DatatypeID      = 0;
  block.Dwork(20).Complexity      = 'Real';
  block.Dwork(20).UsedAsDiscState = true;

  block.Dwork(21).Name = 'CounterStates';%Counter to track the states elapsed
  block.Dwork(21).Dimensions      = 1;
  block.Dwork(21).DatatypeID      = 0;
  block.Dwork(21).Complexity      = 'Real';
  block.Dwork(21).UsedAsDiscState = true;

  for gh=1:block.DialogPrm(2).Data
   if gh<10
     block.Dwork(gh+21).Name = strcat('LUT_0',num2str(gh)); %Look up table
   else
     block.Dwork(gh+21).Name = strcat('LUT_',num2str(gh)); %Look up table
   end
   block.Dwork(gh+21).Dimensions      = block.DialogPrm(1).Data;
   block.Dwork(gh+21).DatatypeID      = 0;
   block.Dwork(gh+21).Complexity      = 'Real';
   block.Dwork(gh+21).UsedAsDiscState = true;
  end
%endfunction


function Terminate(block)
%% Saving LUT to the file

 fid=fopen('LUT.txt','wt+');
  for act=1+21:block.DialogPrm(2).Data+21
    for stat=1:block.DialogPrm(1).Data
      fprintf(fid,'%d\t', block.Dwork(act).Data(stat));
    end;
    fprintf(fid,'\n');
  end
 fclose(fid);
%endfunction


function InitConditions(block)
%% Initilize Mapping

   %NbrProd=index; %number of items for (PM,RM)
   NbrPrCell=5; %Nbr of Prouction cells in model
   NbrPM=block.DialogPrm(3).Data;

   index=0;
   for pmind=1:NbrPM
    for rmind=1:NbrPM-pmind+1
      index=index+1;
      ProdMapp(rmind, pmind)=index;
    end
   end

 %Establishing nbr of combinations for distributors, IS BETTER IF THE
```

```
 %MATRIX IS CREATED FIRST AND THE LENGTH IS RETRIEVED

  NbrCombDist1=5;%Nbr combinations for Distr 1 CHECK ALSO INITIALIZE FUNCTION
IF CHANGED
  NbrCombDist2=5;%Nbr combinations for Distr 2 CHECK ALSO INITIALIZE FUNCTION
IF CHANGED

%  %Initialize (LUT)
%   mat=zeros(block.DialogPrm(1).Data,1);
%   for act=1+21:block.DialogPrm(2).Data+21
%     mat= block.Dwork(act).Data;
%     for stat=1:block.DialogPrm(1).Data
%       mat(stat) = rand(1,1);
%     end;
%     block.Dwork(act).Data=mat;
%   end
%  %end of initialize LUT


 %Load (LUT)
 fid=fopen('LUT.txt','r');
  for act=1+21:block.DialogPrm(2).Data+21 % num of actions
    for stat=1:block.DialogPrm(1).Data % num of states
      fscanf(fid,'%d\t', block.Dwork(act).Data(stat));
    end;
    fprintf(fid,'\n');
  end
 fclose(fid);
%end of load LUT



  %vector=load('LUT.txt'); %load txt file
  %%directly to a vector instead of filling with zeros
  %vector=load('LUT.txt');

 %Pass the information to the Dworks vectors to use them as global variables
  block.Dwork(1).Data=block.DialogPrm(1).Data;
  block.Dwork(2).Data=0; %Start discharged patients in zero
  block.Dwork(3).Data(1)=0; %Initialize state n-1 with invalid state to be
skipped  Recall this structure is a vector lenght 2 [state|action]
  block.Dwork(4).Data(1)=0; %Initialize state n with invalid state to be
skipped    Recall this structure is a vector lenght 2 [state|action]

  block.Dwork(3).Data(2)=0; %Initialize action n-1 with 0
  block.Dwork(4).Data(2)=0; %Initialize action n   with 0

  block.Dwork(5).Data=0.5; %Establishing learning rate, alpha
  block.Dwork(6).Data=0.7; %Establishing discount factor, gamma

  block.Dwork(7).Data=5;%Nbr combinations for Distr 1 CHECK ALSO SETUP
FUNCTION IF CHANGED
  block.Dwork(8).Data=5;%Nbr combinations for Distr 2 CHECK ALSO SETUP
FUNCTION IF CHANGED
```

```
  block.Dwork(9).Data=11;  %First combination for Distr1 is allocated in
Dwork 11
  block.Dwork(10).Data=16; %First combination for Distr2 is allocated in
Dwork 16

  block.Dwork(11).Data=  [32.25, 16.12, 51.60, 0.03];%1 combination ratios
for Distr 1
  block.Dwork(12).Data=  [64.43, 32.22, 0, 0.13];%2 combination ratios for
Distr 1
  block.Dwork(13).Data=  [33, 33, 33, 1];%3 combination ratios for Distr 1
  block.Dwork(14).Data=  [50, 0, 49, 0.06];%[50, 40,9, 0.05];%4 combination
ratios for Distr 1
  block.Dwork(15).Data=  [10, 40, 49.95, 0.05];%5 combination ratios for
Distr 1

  block.Dwork(16).Data=  [23.08, 15.38, 61.54]; %1 combination ratios for
Distr 2
  block.Dwork(17).Data=  [15.38, 10.26, 0]; %2 combination ratios for Distr 2
  block.Dwork(18).Data=  [69.23, 30.77, 0]; %3 combination ratios for Distr 2
  block.Dwork(19).Data=  [33.33, 33.33, 33.33]; %4 combination ratios for
Distr 2
  block.Dwork(20).Data=  [10, 40, 50]; %5 combination ratios for Distr 2

  block.Dwork(21).Data=0; %counter to check going to learn function

%endfunction


function SetInputPortSamplingMode(block, idx, fd)
 block.InputPort(idx).SamplingMode = fd;


for i=1:block.NumOutputPorts
   block.OutputPort(i).SamplingMode = fd;
end
%endfunction

% function SetInpPortDims(block, idx, di)
%   block.InputPort(idx).Dimensions = di;
%   block.InputPort(idx).Dimensions = di;
%
%
% %endfunction

function Output(block)

 NbrPM=block.DialogPrm(3).Data;

 RM1=block.InputPort(2).Data;
 RM2=block.InputPort(4).Data;
 RM3=block.InputPort(6).Data;
 RM4=block.InputPort(8).Data;
 RM5=block.InputPort(10).Data;
```

```
 PM1=block.InputPort(1).Data;
 PM2=block.InputPort(3).Data;
 PM3=block.InputPort(5).Data;
 PM4=block.InputPort(7).Data;
 PM5=block.InputPort(9).Data;

 [PROD1] = Product_index(PM1,RM1,NbrPM);
 [PROD2] = Product_index(PM2,RM2,NbrPM);
 [PROD3] = Product_index(PM3,RM3,NbrPM);
 [PROD4] = Product_index(PM4,RM4,NbrPM);
 [PROD5] = Product_index(PM5,RM5,NbrPM);

 learning=block.Dwork(5).Data;%retrieving learning rate
 discount=block.Dwork(6).Data;%retrieving discount factor

 PrvSt=block.Dwork(3).Data; %Retrieving the previous visited state
 CurreSt=block.Dwork(4).Data; %Retrieving the current chosen state

 if block.InputPort(11).Data == block.Dwork(2).Data % if the input disc rate
equals the stored disc rate
    reward = block.InputPort(11).Data;
 else
     reward = block.InputPort(11).Data;%reward = (block.InputPort(11).Data -
block.Dwork(2).Data)*1.5;
 end

 if (PrvSt(1)>=1 && CurreSt(1)>=1 && block.InputPort(12).Data>0) %just
compute q-fuction if both n-1 and n-2 states are valid and waiting area is
not empty
  block.Dwork(PrvSt(2)+21).Data(PrvSt(1))=
block.Dwork(PrvSt(2)+21).Data(PrvSt(1))+learning*(reward+discount*block.Dwork
(CurreSt(2)+21).Data(CurreSt(1))-block.Dwork(PrvSt(2)+21).Data(PrvSt(1)));
  %vector(PrvSt)=vector(PrvSt)+learning*(reward+discount*vector(CurreSt)-
vector(PrvSt));
 end

% Calculate bounds for potential states based in static ProdMs
 if(PROD1>=1 && PROD1<=5 && PROD2>=1 && PROD2<=5 && PROD3>=1 && PROD3<=5 &&
PROD4>=1 && PROD4<=5 && PROD5>=1 && PROD5<=5) %if state is valid %[to cover
the whole possible states ----> PROD?<=15 (for all PROD?'s]
  Maxm=-1E400;
  Distr1index=1;
  Distr2index=1;

  block.Dwork(21).Data=block.Dwork(21).Data+1;

  [state]=Index_state(PROD1,PROD2,PROD3,PROD4,PROD5); %sense the state
  block.OutputPort(7).Data = state;

  if  rem(block.Dwork(21).Data,144)==0  %going for explore movement every 180
timesteps

     %Generate 100 values from the uniform distribution on the interval [a,
b].
```

```
    %r = a + (b-a).*rand(100,1);

    Distr1index= round(1 + (5-1).*rand(1,1))  ;%random value between 1 & 5
    Distr2index= round(1 + (5-1).*rand(1,1))  ;%random value between 1 & 3

 else %Greedy movement %look for the highest Q-Value



     for hh=1:block.Dwork(7).Data % num of elec choices 1-5
        for ii=1:block.Dwork(8).Data % num of water choices 1-5
           [action_ind] = Index_Action(hh,ii);
           %Index_Action(hh,ii)
           %block.Dwork(21+action_ind).Data(state)
           if block.Dwork(21+action_ind).Data(state) > Maxm   %Keep in mind
the offset with the rest of the Dworks 19
             Maxm=block.Dwork(21+action_ind).Data(state);
             Distr1index=hh;
             Distr2index=ii;
           end

        end
     end
    Distr1index
    Distr2index
 end


%  else
%   Distr1index=1; %To force the agent to choose 1st state if PMs are outta
bounds
%   Distr2index=1; %To force the agent to choose 1st state if PMs are outta
bounds
%   state=1; %To force the agent to choose 1st state if PMs are outta bounds
 end;

 [action_ind] = Index_Action(Distr1index,Distr2index);

 block.OutputPort(5).Data =PrvSt;   %PrvSt;
 block.OutputPort(6).Data =CurreSt;

 PrvSt=CurreSt;          %updating previous state  for next time step
 CurreSt(1)=state;       %updating current  state  for next time step
 CurreSt(2)=action_ind;  %updating current  action for next time step

 block.OutputPort(3).Data=reward;
 block.Dwork(2).Data = block.InputPort(11).Data; %storing discharge patients
to have historical values in order to generate Rewards

 block.OutputPort(1).Data= block.Dwork(block.Dwork(9).Data-1
+Distr1index).Data; %Output to distributor 1
 block.OutputPort(2).Data= block.Dwork(block.Dwork(10).Data-
1+Distr2index).Data; %Output to distributor 2
```

```
%  matrix=zeros(block.DialogPrm(2).Data, block.DialogPrm(1).Data);
%
%  for inde=1:block.DialogPrm(2).Data
%    matrix(inde,:)=block.Dwork(19+inde).Data; %Keep in mind the offset with
the rest of the Dworks 19
%  end
%
%
%
 %block.OutputPort(4).Data =
matrix; %block.Dwork(2).Data;%RM4;%2*block.InputPort(4).Data;

 block.Dwork(3).Data=PrvSt; %Storing the previous visited state
 block.Dwork(4).Data=CurreSt; %Storing the current chosen state

%endfunction

 function [ind] = Product_index(PM,RM,NbrPM)
 ind=(PM-1)*NbrPM+RM;
 switch PM
    case 3
      ind=ind-1;
    case 4
      ind=ind-3;
    case 5
      ind=ind-6;
 end
%End function productivity index

function [state_ind] = Index_state(Pro1,Pro2,Pro3,Pro4,Pro5)

  state_ind=(Pro1-1)*5^4+(Pro2-1)*5^3+(Pro3-1)*5^2+(Pro4-1)*5+Pro5; %[to
cover the whole possible states ----> state_ind=(Pro1-1)*15^3+(Pro2-
1)*15^2+(Pro3-1)*15+Pro4]

  %End function state indexing

function [action_ind] = Index_Action(Act1,Act2)
action_ind=(Act1-1)*5+Act2;
%End function action indexing
```