A Framework for Resource Allocation in Time Critical Dynamic Environments Based on Social Welfare and Local Search and its Application to Healthcare

by

Dean Shaft

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis provides an artificial intelligence approach for the problem of resource allocation in time-critical dynamic environments. Motivated by healthcare scenarios such as mass casualty incidents, we are concerned with making effective decisions about allocating to patients the limited resources of ambulances, doctors and other medical staff members, in real-time, under changing circumstances. We cover two distinct stages: the Ambulance stage (at the location of the incident) and the Hospital stage (where the patient requires treatment). Our work addresses both determining the best allocation and supporting decision making (for medical staff to explore possible options). Our approach uses local search with social welfare functions in order to find the best allocations, making use of a centralized tracking of patients and resources. We also clarify how sensing can assist in updating the central system with new information. A key concept in our solution is that of a policy that attempts to minimize cost and maximize utility. To confirm the value of our approach, we present a series of detailed simulations of ambulance and hospital scenarios, and compare algorithms with competing principles of allocation (e.g. sickest first) and societal preferences (e.g. egalitarian allotment). In all, we offer a novel direction for resource allocation that is principled and that offers quantifiable feedback for professionals who are engaged in making resource allocation decisions.

Acknowledgements

I would like to thank my supervisor, Professor Robin Cohen, for her guidance and support. I appreciate the great amount of time and effort that Robin has devoted to my research throughout my time at the University of Waterloo. I would also like to thank my readers, Professor Peter Van Beek and Professor Jesse Hoey, for their comments and useful feedback. I am grateful for the efforts of my committee.

I acknowledge that financial support was received from NSERC (the Natural Sciences and Engineering Council of Canada) and from its Strategic Network hSITE (Healthcare Support through Information Technology Enhancements), The University of Waterloo, and Cheriton School of Computer Science.

Thank you to my colleagues in the AI group: John Doucette, Hadi Hosseini, Graham Pinhey, and Noel Sardana. I enjoyed both our academic and non-academic discussions.

My time as a graduate student would not have been the same without the people that I have met at The University of Waterloo. I would like to thank all the friends that I have made here for their support. A special thanks (in no particular order) to Valerie Sugarman, Oliver Trujillo, Cecylia Bocovich, Stephen Kiazyk, Hella Hoffmann, Jack Thomas, Rhiannon Rose, Aaron Moss, and Marianna Rapoport,

Most importantly, I would like to thank my family for their endless love and support. Thank you to my parents, Tali and Eli, my younger brother, Ben, my older brother, Niv, and his wife, Robyn.

Dedication

This thesis is dedicated to my parents, Tali and Eli Shaft.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Within the field of artificial intelligence, an intelligent agent represents an entity that perceives and acts in an environment rationally in order to reach a certain goals or perform certain tasks; multiagent systems represents the system where these intelligent agents interact with each other [45]. Multiagent resource allocation takes resources and agents as input and outputs how the resources should be allocated to the agents. This thesis concerns multiagent resource allocation and proposes a framework for performing this task in dynamically changing environments where real-time decisions are required.

Chevaleyre et al.'s survey of multiagent resource allocation (MARA) systems distinguishes between distributed and centralized approaches [8]. In this thesis, we propose a centralized solution, one that uses local search in order to discover preferred allocations. Chevaleyre et al.'s discussion of MARA clarifies that agents in need of resources will have preferences over what they receive; they also explain that determining the value of an allocation can be achieved by virtue of some kind of global utility function.

In this thesis, we are interested in a particular kind of environment, one that can be characterized as follows: i) it has indivisible resources ii) the resources may either be shareable or unshareable iii) the multiagent system is cooperative; as a result agents are not attempting to deceive their peers iv) there is a very general characterization of agent preferences, which applies to all of the agents in the system (they all want to receive the best resource that improves their utility) but there may still be differing needs, amongst agents (for example, in a healthcare domain one agent may need a heart surgeon whereas another requires a brain surgeon) v) the preferences of users are in essence dictated by their current state: what agents in this state would require as resources vi) the environment is one in which the cost of a poor allocation is important to be modeling; this means that instead of introducing a utility function to maximize, we are more focused on reasoning with a cost function, to minimize vii) there is the challenge of coping with agents in need of resources for whom no resources have in fact been allocated viii) the environment is dynamically changing and the allocation solution must be computed in real-time.

The design that we will propose in order to provide for resource allocation in environments such as these is characterized as one of solving a constraint satisfaction problem, cast in terms of conducting a local search towards a solution. In Chapter 2 we elaborate on what needs to be specified when adopting this particular approach (including preferred search strategy, heuristic for selecting neighbouring solutions, options for restarting the search, etc.). While viewing the solution of constraint satisfaction under dynamic conditions as local search is not a novel depiction (see for instance [42]) what we are now proposing and illustrating is using this kind of artificial intelligence paradigm as the basis for resolving resource allocations.

Since MARA problems require some kind of function to compare alternative solutions, what we outline in this thesis is an approach for clarifying the cost functions that will enable the allocation options to be compared. Towards this end, we propose that social welfare functions be used as the central metric. Chevaleyre et al. discuss the role that social welfare functions have played in the design of MARA systems; he also provides a nice survey of the different kinds of functions which may be introduced [8]. This discussion depicts these functions as useful in reflecting the preferences of agents, towards improved utility in environments such as auctions, with negotiation. In our system, as will be shown, we make the decision to employ social welfare functions in order to compare the differing costs of solutions. It will turn out to be the case that this metric is especially valuable for the kinds of applications in which we are interested.

We have already characterized the kinds of environments we wish to model and in which our MARA proposal will operate. One particular application is that of healthcare. We were in fact motivated by trying to address how to assign ambulances to hospitals, especially when faced with critical, burdensome scenarios such as mass casualty incidents. But we wanted our solution to be sufficiently general that it could also handle the second phase of MARA within this context: assigning doctors to patients upon arrival at the hospital. We also imagined that our framework would be of value in other environments, such as allocating fire fighters to a new, massive fire (which would require the same kind of system characterization). Faced with a MARA challenge in a context like this, it occurred to us that what would also be important to offer as part of our design are additional elements: i) an articulation of the sensing that would occur in the environment that would enable the updating of parameter values ii) a characterization of when it would be possible to change the proposed allocation of resources (e.g. certainly before the actions to be performed by the resource have begun, a kind of buffering situation) iii) support for decision making by the humans who are interested in learning the proposed allocation of resources.

This then caused us to ensure, when designing our framework, that it could be implemented, thrown into simulated scenarios, in such a way that various graphs could be generated which would both i) confirm the value of our particular design, in comparison with less reasoned approaches to MARA ii) provide output of value to decision makers, in order for them to adjust certain parameter settings, in an effort to determine their best decision making strategies.

This thesis devotes considerable effort to clarifying what decision makers can specify and can see outcomes for, as part of our decision making support within our overall framework. As will be explained in greater detail in Chapter 3, we allow for variation of i) preferred social welfare functions ii) preferred principles of allocation (e.g. sickest first vs. first come first served) iii) experimenting with different buffering times (for how long one can adjust the proposed initial allocation before it is locked in and no longer adjustable (the gains in solving quickly vs. the losses in ending up with less desirable allocations) iv) the effect of setting differing times for when to sense and thus to update the parameter values.

In the chapters that follow we provide background information including competing approaches and related work that serves to inform our solution (Chapter 2), a description of our proposed model in detail, including its central algorithms (Chapter 3), a depiction of this model for the application of healthcare, outlining proposed parameters to model and their range of possible values, as well as useful domain-specific cost functions (Chapter 4), and a validation of this model in this specific context (Chapter 5) providing not only graphs which show the performance of our approach but also sample output that can be presented to decision makers in this context. In Chapter 6 we return to summarize our contributions and to discuss future directions.

In all, we are offering a clear characterization of local search as a method for MARA, proposing the use of social welfare functions as the basis of computing and comparing costs. We illustrate the use of this framework in detail for the application of healthcare, highlighting what our design makes possible for decision makers. Through simulations, we are able to illustrate the effectiveness of the framework and thus its potential contribution for healthcare applications. Our final discussion clarifies the central components our design that makes it amenable to a variety of possible contexts, outlining as well what distinguishes it from other approaches.

Chapter 2
Background

In this chapter, in order to determine the requirements for our solution, we include a literature survey for some background information. Areas that will be covered include patient scheduling, decision support systems, resource allocation, and constraint satisfaction. Although there is overlap in the topics between the literatures, the sections have been divided into subtopics for ease of reading. Once we have presented our proposed approaches in detail (Chapters 3-5), we will return to provide additional compare and contrast (Section 6.2).

2.1 Patient Scheduling

Scheduling patients in a hospital can be carried out from many different approaches. These approaches include appointment exchanges [43], coordination mechanisms [13, 21], genetic algorithms [37, 49], and multi-agents [13, 21, 31, 33, 43], where some of them [13, 21, 43] are a combination of approaches. Simulations and experimentations are typically performed to show the benefit of the approach.

Vermeulen et al.'s [43] paper discusses patient scheduling in a hospital that uses a multi-agent Pareto-improvement appointment exchanging algorithm (MPAEX) with the objective of minimizing completion times of all their patients. There are a few techniques used to reach their objective. The first is with the multi-agent system, where agents are either resource agents or patient agents. A resource agent represents medical professionals or medical equipment (e.g. examination room). In addition, resource agents include their constraints and their preferences. A patient agent simply represents the patient in the hospital and includes their needs as well as their preferences. Patient agents will have timeslots scheduled with the resource agents so that they can be treated. Scheduling the timeslots applies Pareto improvement technique, where a patient will be scheduled to a timeslot only if no other patient will become worse by that scheduling. It is important to note that activities performed in a hospital each has its own duration and each agent can only be part of one activity at a time. Moreover, the dynamic nature of hospitals causes activities to be added continuously depending on the arrival and departure of patients as well as resources. All these factors are taken into account using the MPAEX algorithm. Using the technique of appointment exchanging between agents, patients are initially assigned timeslots by corresponding with resource agents then the patient agents will continuously reschedule until exchanges can no longer be made.

Decker and Li's [13] work also has a multi-agent solution but they use a Generalized Partial Global Planning (GPGP) approach. In this paper, agents represent the ancillary and nursing units of the hospital while the resources represent the patients. The units perform different tasks (e.g. blood test) on the patients. These tasks have a few considerations. The first is that tasks can have relationships between them, for example, completing a certain task before others. Next, tasks should not be redundant. Finally, depending on the task, a patient can be considered a non-shareable resource meaning it will not be able to be in two different ancillary units at the same time. The authors mention that cooperative agents are required in order to coordinate the tasks. Furthermore, agents will have comparable utility measures when determining the schedule. Schedules are creating through a bidding process. Each agent produces a local schedule and they bid for the time interval to complete their task. If the agent loses the bid then it will reschedule. If the agent wins the bid then it will execute the task. However, the winning agent may still be required to reschedule due to the dynamic nature of scheduling in a hospital. One of the reasons identified in this paper for scheduling is to minimize time for setup, which will minimize the amount of time a patient needs to stay.

Coordination of multi-agent patient scheduling is similarly found in Kanaga, Darius, and Valarmathi's [21] paper. This work has an auction based mechanism with three types of agents. The Common Agent registers the patient, sets the initial plan for treatment, and assigns a priority to the patient based on their health. Examples of a Common Agent include general physicians and hospital receptionists. The Resource Agent can be doctors, specialists, and hospital equipment (e.g. MRI machine). The doctors and specialists have their own beliefs, goals, availability, and preferences. The Patient Agent represents a patient with a treatment plan consisting of tasks to be completed at a certain time. Each patient will have preferences for resources and time slots. The patient's health and priority level may change over their hospital visit. The authors describe an auction based mechanism where each Resource Agent is an auctioneer and each Patient Agent is a bidder. In order for a Patient Agent to minimize their waiting time they places bids to the Resource Agent for time slots. Once the auction is over, the bidders are informed who won the time slot. If a Patient Agent lost the auction and still needs the unavailable resource then the Resource Agent will trigger an auction when a time slot becomes available. Again, minimizing the patient's waiting time will benefit their recovery.

Similar to the other work, Paulussen et al. [33] discusses patients may needing to go through many different wards and units in the hospital to be able to receive examinations and treatments. A patients schedule in the hospital may change based on the changes to their health. Furthermore, schedules can change due to tasks being added from being required or tasks being removed since they became obsolete. Resource and patient agents are mentioned for the multi-agent based approach. Patient agents will compete against each other for the limited resources in the hospital. Moreover, patient agents have a cost function that measures their health state progress, which takes into account the development of their health state. The patient's goal is to be treated so that their health state increases. An opportunity cost arises from delaying treatment to a patient. Resource agents also have a cost function but it is measured in monetary costs. The paper identifies the goals of minimizing a patient's stay at the hospital and minimizing the idle times for resources. To achieve this goal, a distributed approach is used to negotiate and reschedule. Patients try to decrease their opportunity costs by improving their initial schedule through negotiations. During the negotiation process, a patient agent will use a resource agent in order to try to obtain an earlier time slot from the other patient agents. Rescheduling of the time slots will occur only if the expected gains exceed the costs.

In contrast to the previous papers, the work of Xiao, Osterweil, and Wang [49] use an evolutionary algorithm approach to schedule resources in the emergency department of a hospital. Patients require the hospital resources for treatment but these resources are limited. The authors discuss using an incremental resource scheduling method for rescheduling tasks that occur in a specific time-window. Finding a window that is the right size is important in order to avoid frequent rescheduling, high cost, and low accuracy. The paper identifies three types of scheduling constraints for an emergency department. These constraints are capability, availability, and step execution order. Certain components of the framework are described by the authors. The scheduler component relies on a genetic algorithm to construct the schedule. Rescheduling is determined by the rescheduling indicator component. The scheduling activity set constructor contains all the information needed for schedule decision making including activities that may occur in the near future. The request to perform an activity contains the type of resource required, the capability, and the minimum skill level. The resources are description includes their availability, capability, skill level, and productivity. The final component of the framework is the system execution component, which comprises of information needed for updates.

The work of Priya, Anandhakumar, and Maheswari [37] also has a genetic algorithm for scheduling. The authors discuss using Radio Frequency Identification (RFID) tags on hospital equipment, medical staff, and patients to track where they currently are in the hospital. Moreover, their system will determine availability and whether they require service. When a patient arrives at the hospital, they will be given a wristband that contains an RFID tag. Each patient's RFID tag will store information about the patient including their medical files. The RFID system will use the information about the medical staff and equipment in order to schedule them to a patient. The RFID system will schedule patients so that the time between arrival and beginning of treatment will be minimized. The paper acknowledges that multiple patients can arrive at the hospital at the same time, which changes the objective to be minimizing the average set up time of these patients. Moreover, the system will schedule the appointment slots for each doctor to be continuous as to maximize utilization. The authors propose a dynamic scheduler using Nondominated Sorting Genetic Algorithm II (NSGA-II), an evolutionary algorithm, to find the optimal schedule. The algorithm ensures that a time slot will not be allocated to more than one patient and that the allocation will provide the patient with the appropriate resource. Scheduling in hospitals is discussed in Niemann and Eymann's [31] paper. The approach is agent based with decision support systems. It will be covered in the following subsection.

Summary: Various design decisions in this thesis are in agreement with the modeling performed in several of the references of this subsection, in particular having patients represented by agents and tracking central features of resources. We return to contrast our solution with these approaches in Section 6.2.

2.2 Decision Support Systems

Combining multi agent systems with decision support systems for scheduling in hospitals is included in the paper by Niemann and Eymann [31]. The authors discuss a hybrid approach, which allows a human to make the final decision based on the information provided by the fully automated system. The multi agent part of their approach uses active (e.g. medical staff and patients) and passive (e.g. medical equipment and rooms) software agents. It is necessary for these agents to be able to communicate with each other. The paper suggests personal digital assistants for a two-way communication between them. This informs agents about changes to schedules as well as the ability to delegate decisions about the schedule. Sensors and effectors are used to link the logical and real world. Rather than having each human agent manually logging their activities, the authors propose using either Radio Frequency Identification (RFID) or Wireless Local Area Network (WLAN) to monitor them. The system will infer whether the agent is active or idle. If the agent is active then it is cannot make another appointment. However, if the agent is idle then it is available for an appointment. Inferring the location of the agents is appropriate for the system since it only provides suggestions and a human will be making the final decision. The paper mentions a layered architecture for scheduling. The bottom layer is called identification, which receives all the data from the incoming signals and sends appropriate information to the next layer. The context detection layer infers the position of the agent. Following this layer, the integration layer obtains the schedule and checks the validity by comparing the agent's position with the position determined by the previous layer. If all the agents agree on the schedule then it is valid. However, if one of the agents determines that the schedule is invalid then they must send the information to the optimization layer. The optimization layer is the top layer and is responsible for fixing the schedule. The software agent will attempt to create a valid schedule that still follows the previous constraints. If the software agent cannot create a valid schedule and needs medical knowledge then the human agent will be responsible for making the schedule. The authors mention that the advantages of this hybrid system are that the automated system is able to check schedule validity quickly and only a human can analyze the entire situation.

Taboada et al.'s work [40] discusses a decision support system for hospital emergency departments. The project will help directors of the emergency department make decisions about efficient resource use. The authors' work includes an agent based model and simulation. Agents are either active (e.g. patients and doctors) or passive (e.g. IT infrastructure or laboratories). State machines are used to model the agents, where a state is a collection of state variables. Depending on the kind of agent, state variables could include symptoms, location, level of experience, or communication skills. An agent can communicate to an individual or a group. The authors mention the various environments in the model, which includes admission, waiting rooms, and treatment zones. There are five levels to determine the priority given to the patients during the triage process, where level I is the highest and level V is the lowest. The paper describes that the simulations as well as the validation and verification of the model was done with the participation of a team of emergency department staff from the Hospital of Sabadell (Spain). The authors conclude the paper with experimental results from the simulation that showed a decrease of patient length of stay in the emergency department and a decrease in required number of physicians.

The paper of Wong, O'Hare, and Sallis [48] use the critical decision method (CDM) to elicit information from ambulance dispatchers in Australia. Five ambulance dispatchers were interviewed with the intention of understanding their process and the appropriate information for decision support. Five goal states emerged after the interviews were transcribed, summarized, and analyzed. The goal states are notification of emergency, maintain situation awareness, planning resource to task compatibility, speedy response and maintain history of development. Each goal state provided guidance on decision support in the naturalistic decision making (NDM) environments.

Heimly and Nytro's [18] work provides clinical guidelines for referrals given by primary care. Their study transpired in a Norwegian hospital using general practitioners with the intention of improving the quality of the referrals. These were electronic referrals that the general practitioners have already been using in their practices. When sending the patient to a specialist, the guidelines included in the referrals provided decision support. It is important to note that the general practitioner can choose whether or not to follow these guidelines. The paper identifies that the guidelines also helped the general practitioners determine whether or not a patient will be referred to a specialist.

Summary: Our approach of considering sensing to inform resource allocation and supporting decision making by medical professionals fits well with the perspective of these authors. In Section 6.2, we offer additional comparisons with this work.

2.3 Mass Casualty Incidents

Mass casualty incidents in the health care environment have been dealt with in previous papers. Doucette's work [14, 15] described a hospital that uses an automatic scheduling system that would allocate resources, such as a doctor, to a patient. The strategy is to use preemption in order to meet the patient's requirements. For example, a patient would need to be seen by a doctor within the first two time cycles. This could be described as a constraint satisfaction problem by having the patients as the variables and the doctors as the value in the domain.

Branas, Sing, and Perron's work [5] also discusses mass casualty incidents. They focused on mass casualty incidents that occurred during a three year period in Maryland. Eight cases were described in that paper with various numbers of injured individuals, ambulance vehicles used, and hospitals involved. Their paper included some helpful knowledge that was applied to the example in this paper, including number of victims and hospitals. Moreover, Einav et al.'s work [16] provided other kinds of information regarding a mass casualty incident. Their work involved thirty three mass casualty incidents that occurred during a two year period in Israel. Unlike the work in [5], Einav et al.'s work [16] is less descriptive in each event that occurred but rather provides different kinds of statistical categories, including initial time for the ambulance and distance to the hospital, that were applied to the example of this paper.

In the literature regarding a health care environment, the measurement that describes the health of a person greatly varies between papers. Einav et al.'s work [16] took a discrete approach by stating that the victim's health is urgent, not urgent, or unknown. Work by Feeny et al. [17] had a continuous approach to a patient's health. As seen in Figure 2.1, which shows Table 3 from Feeny et al.'s work [17], the health is represented by a utility function where perfect health is at 1.00 to dead at 0.00. The utility function consumes a value from each attribute, where each level is used to describe the patient's health. The closer the value is to level 1, the better the individual's health. For instance, the ability to see well from close and far distances would be a level 1 for the vision attribute and not being able to see at all would be a level 6. Patients with the same type of injury may have different health values based on less concrete attributes like emotion.

These approaches have their strengths as well as weaknesses. For this example, discrete values are needed but the health of the individual should be known and should be categorized in a few health states. The American Hospital Association's work [1] has described four health states, which can be seen in Figure 2.2. In ascending severity order, they are good, fair, serious and critical. This work is in regards to releasing information about a patient's condition, therefore it is also simple and can be easily understood by those not in the health field.

Once the health of an individual is known, there are many factors to be considered when assigning a hospital to them. Østerdal's work [32] states two interesting questions about these assignments. The first is the idea of a health state that is worse than being dead. The second deals with discriminating against age and whether it is valid. I did not deal with these issues in this paper but they can be addressed in future work. Weale's work [44] also raised an interesting consideration about assignments. Weale discussed the problem of equity to determine whether assignments should be based on clinical need or productivity. When dealing with fairness, it is important to take a global view to determine whether a patient should be sent to a certain hospital.

TABLE 3. HUI3 Multi-Attribute Utility Function: Simplified Format on Dead-Perfect Health Scale

Vision		Hearing		Speech		Ambulation		Dexterity		Emotion		Cognition*		Pain	
x_1	b_1	x_2	b_2	x_3	b_3	x_4	b_4	x_5	b_5	x_6	b_6	x_7	b_7	x_8	b_8
1	1.00	1	1.00	1	1.00	1	1.00	1	1.00	1	1.00	1	1.00	1	1.00
2	0.98	2	0.95	2	0.94	2	0.93	2	0.95	2	0.95	2	0.92	2	0.96
3	0.89	3	0.89	3	0.89	3	0.86	3	0.88	3	0.85	3	0.95	3	0.90
4	0.84	4	0.80	4	0.81	4	0.73	4	0.76	4	0.64	4	0.83	4	0.77
5	0.75	5	0.74	5	0.68	5	0.65	5	0.65	5	0.46	5	0.60	5	0.55
6	0.61	6	0.61	6	n/a	6	0.58	6	0.56	6	n/a	6	0.42	6	n/a

Formula (Dead-Perfect Health Scale)

$$u^* = 1.371\ (b_1{}^*b_2{}^*b_3{}^*b_4{}^*b_5{}^*b_6{}^*b_7{}^*b_8) - 0.371$$

where u* is the utility of a chronic health state† on the utility scale where dead‡ has a utility of 0.00, and healthy† has a utility of 1.00.

*The single-attribute utility score for Level 3 Cognition is greater than the single-attribute utility score for Level 2 Cognition.
†Chronic states, and the perfect health state, are here defined as lasting for a lifetime.
‡Dead is defined as immediate.

Figure 2.1 – Table 3 from [17]

Good - Vital signs are stable and within normal limits. Patient is conscious and comfortable. Indicators are excellent.

Fair - Vital signs are stable and within normal limits. Patient is conscious, but may be uncomfortable. Indicators are favorable.

Serious - Vital signs may be unstable and not within normal limits. Patient is acutely ill. Indicators are questionable.

Critical - Vital signs are unstable and not within normal limits. Patient may be unconscious. Indicators are unfavorable.

Figure 2.2 – Health conditions from [1]

So far the papers have only dealt with the injured individual in regards to the assignments. Chevaleyre et al.'s work [8] focuses on social welfare to give a global view of the scenario. This will be discussed in the next section.

Summary: As in these papers, in our work we model patient health states in order to direct our resource allocation decisions. The approach that we use and its rationale are covered in Chapter 3-5.

2.4 Resource Allocation

Allocating resources can cover different domains and techniques. Some of the previous papers, including Priya, Anandhakumar, and Maheswari's work [37], identified their scheduler as a solution to the resource allocation problem. Meuleau et al.'s paper [27] solves resource allocation for planes by modeling the problem as a Markov decision process (MDP). Weng et al.'s [46] work covers resource allocation in a hospital's emergency room department. The authors explore using linear along with nonlinear pricing models and attempt to solve this problem by maximizing in terms of profit.

Chevaleyre et al.'s [8] work provides as a survey resource allocation. The paper describes the types of resources. Resources can be discrete (e.g. CT scanner) or continuous (e.g. energy to power a hospital). Some resources can be divided (e.g. cotton swabs) and others are indivisible (e.g. electrocardiography machine). In addition, resources should specify if they are shareable or not. For example, recovery rooms can be used by multiple patients or can be private, which makes it usable to only one patient and therefore will not be shareable. It is important to note that resources may change over time. Finally, resources should identify whether they are representing a single- (e.g. nurses) or multi- (e.g. bandages) unit setting. The paper also discusses allocation. A centralized approach to allocation has the final decision made by a single entity. In contrast, a distributed approach determines the allocation through local negotiations. Allocations can be decided by auction protocols or negotiation protocols. During negotiation, the agreement can be viewed from a local or global perspective. The local view only focuses on the individuals in the negotiations. The global view evaluates the entire allocation and the term social welfare is used a metric to determine the quality of the allocation. The allocations in social welfare depend on efficiency and fairness. An efficient allocation can be achieved with a technique called Pareto optimality, where it is impossible to improve an individual's allocation without worsening others. A fair allocation can be achieved with a technique called envy-free, where all the individuals are satisfied with their own allocation and would not prefer any other individual's allocation. Envy can be minimized by reducing the number of individuals who are envious or reducing the amount of envy the individuals have. Social welfare function relies on each individual's utility to determine the social welfare. Each individual tries to maximize their utility and the social welfare is also intended to be maximized. An allocation is determined to be better than another by comparing the values determined by the social welfare function. The first social welfare function, seen in (1) is called the **Utilitarian Social Welfare**.

$$SW_u(P) = \sum_{i \in A} Ui(P) \qquad (1)$$

This function sums the all the individual's (A) utility (U) given the allocation (P). Note that this function is not affected by non-positive values for the utility. In contrast, the next social welfare function is greatly affected by non-positive values. **Nash Product**, seen in (2) multiplies all the individual's utility in the allocation.

$$SW_N(P) = \prod_{i \in A} Ui(P) \quad (2)$$

This function works best when all the utility values are non-positive. It is important to note that the value of the social welfare increases as the values of each utility function reach equality, where the large values decrease and the small values increase to the same middle value. Another advantage of using Nash product is that it is scale independent. The third social welfare function is the **Egalitarian Social Welfare**, seen in (3).

$$SW_e(P) = \text{Min } \{Ui(P) \mid i \in A\} \quad (3)$$

This function sets the social welfare as the lowest utility value in the allocation. Here the worst off individual sets the social welfare. In contrast, **Elitist Social Welfare**, seen in (4), sets social welfare as the highest utility value in the allocation.

$$SW_{el}(P) = \text{Max } \{Ui(P) \mid i \in A\} \quad (4)$$

The best off individual sets the social welfare. **Rank dictator**, seen in (5), is the final social welfare function discussed in the paper.

$$SW_k(P) = (v\uparrow_p)_k \qquad (5)$$

The social welfare is set to the k-th smallest utility value. This means that when k is 1 then social welfare is equivalent to Egalitarian Social Welfare and when k is the number of individuals then the social welfare is equivalent to Elitist Social Welfare. Furthermore, the social welfare could be the median rank dictator, which is when k is the rounded up whole number value of the number of individuals divided by two. Each social welfare function has its benefits and the appropriate approach should be used to determine the best possible allocation.

Principles of allocation are discussed in a paper by Persad, Wertheimer, and Emanuel [34] focused on resource allocation for medical applications. There are four categories for eight simple principles. The first category is about treating people equally. **Lottery** and **first-come-first-served (FCFS)** are two principles in this category. The advantage of these principles is the equal opportunity of receiving the resource but the disadvantage is that these principles ignore other relevant factors. The second category of principles favors the worst off. The first principle in this category prioritizes the **sickest first**. An issue with this principle is that future health is not considered since an individual who is not currently sick will be ignored but may become sick in the future. **Youngest first** is another principle in this category. The paper identifies a disadvantage of this prioritization is that it ignores other relevant factors. The next category of principles takes a utilitarian approach by maximizing total benefits. **Saving the most lives** is the first principle in the category and focuses on maximizing the lives saved. **Prognosis** is the second principle in the category and refers to maximizing total life-years, which represents the remaining number of years in one's life. The issue with these principles is that other relevant factors are ignored. The final category promotes and rewards social usefulness. **Instrumental value** allocation will prioritize individuals based on their agreement of future behavior. For example, promising to improve one's health in order to receive the limited resource. On the contrary, **reciprocity** allocation prioritizes based on past behavior. The issue with this final category is that the prioritization is vulnerable to misuse. All eight simple principles have disadvantages but combining the principles may create better allocation systems. The paper identifies four multi-principle allocation systems. The first system is called United Network for Organ Sharing (UNOS) points systems, which combines first-come-first-served, sickest-first, and youngest-first. This system is flexible by allowing a principle to have a greater influence than others. The next allocation system modifies the prognosis principle. Quality-adjusted life-years (QALY) allocation not only considers the quantity of life-years but also considers the quality. Similarly, the third allocation system is disability-adjusted life-years (DALY). Unlike QALY, DALY allocation includes the instrumental value principle. The final multi-principle allocation system is the complete lives system. This system combines youngest-first, prognosis, saving the most lives, lottery, and instrumental value principles. There is no algorithm for this allocation system and the authors mention that it should be thought of more as a framework.

Summary: Social welfare and principles of allocation are central concepts in our proposed model and its validation, described in detail in Chapter 3-5. There are a variety of other approaches for multiagent resource allocation, against which we can compare and contrast our proposed local search approach; Section 6.2.

2.5 Constraint Satisfaction Problems

The constraint satisfaction problem is defined in Bessiere's work [4] as a set of variable with values in the domain and a set of constraints that represents the relationship between the variables and their values. Satisfying all these constraints will result in the solution to a constraint satisfaction problem. In [38], hard constraints are described as constraints that cannot be violated and soft constraints can be violated. When there is soft constraints in a problem then an evaluation function is used to measures its violation and solution tries to minimize these violations. A dynamic constraint satisfaction problem is described in [4], [12], and [42] as a sequence of constraint satisfaction problems, where each subsequent problem represents the restriction, addition of constraints, or relaxation, removal of constraints, based on the changes that occurred from the previous problem. The changes that occur from problem to problem represent new input, which changes the view of the problem and may cause inconsistency with the solution to the new problem [12].

With the problem changing, there will need to be multiple solutions and work by Rossi, van Beek, and Walsh [38] describes three techniques. The first is to reuse aspects of the old solution when computing the new solution. The second is reusing some of the previous reasoning process when generating a new solution. The final technique is preemption, which involves looking in advance for solutions based on likely changes.

There are concerns when finding new solutions that [29], [38], and [42] identify. Minimizing the need to change a solution allows for robust solutions, which means the new solution may only need a minor alteration or none at all. If a modification to the solution is required, it is important to minimize the cost for the modification. Another concern is to minimize the reaction time by obtaining a new solution as quickly as possible. When dealing with these minimizations, it is essential to avoid choices not involved in the current problem, undoing choices not involved in the current problem, and ensure that local changes only affect the current problem. In Mittal and Falkenhainer's work [29], they identify active variables to understand what needs to be changed and what can stay the same.

Local search is discussed in [38] as an approach to solve constraint satisfaction problems. The idea is to start with an initial solution, assignment of values to variables (constraints may be violated). This solution will continuously evaluate neighbor solutions and select a solution that is an improvement until a stopping criterion is reached. Neighbor solution refers to a solution that is a variation to the current solution, which would be a different assignment in the solution. The neighborhood, the neighbors for the current solution, can vary. There are many methods in local search but in our work we focus on hill climbing, which constantly searches for a better solution until it reaches an identifiable peak. Hill climbing will select a new improved solution or it will be the best solution in the neighborhood. There are two approaches called best-improvement and first-improvement. Best-improvement looks through every neighbor in the neighborhood and selects the best solution. First-improvement looks through its neighbors in the neighborhood and selects the first solution that is better than the current solution. As explained in Chapter 3 we adopt the first-improvement option. Local search continues to find better solutions until a terminating criterion is met, such as the best solution is found (all constraints satisfied) or the predetermined limit of number of iterations has been reached. Sometimes local search can reach local optima so to solve this random restarts can be used. Random restart creates a random initial state and repeats the local search algorithm a certain number of times. The best solution of the random restarts is returned as the solution.

Summary: The multiagent resource allocation algorithms which constitute our proposed solution adopt a local search approach for resolving the inherent constraint satisfaction problem. Various terms and concepts clarified in this subsection will be part of our model and its validation. This is all covered in Chapter 3.

2.6 Discussion

Decisions were to be made to determine the necessary approach for our solution and this section briefly discusses these choices. In this work, local search is selected to solve the multiagent resource allocation problem in the emergency department environment. The reasoning for this approach is that we can use local search to make time critical real time decisions about allocating resources to patients.

Moreover, Chevaleyre et al.'s [8] work can be used to create the best possible assignment of hospitals to patients in terms of fairness and efficiency. Selecting a social welfare to assign the resources to the patients depends greatly on the intentions. Each of the social welfare approaches has its own advantages and disadvantages. Utilitarian and Egalitarian social welfare can use any values so they are suitable choice for a global view of a mass casualty incident. Nash product would only be an appropriate social welfare function if there are only positive values. Elitist social welfare does not seem appropriate either because when a mass casualty incident occurs, having one victim with perfect health and everyone else with bad health is not ideal. Moreover, Rank Dictator social welfare function is dependent on the k value selected and would be better left for future work.

Finally, work by Persad, Wertheimer, and Emanuel [34] discuss many different principles of allocation. The simple principles will be used in our work because there are numerous principles that have unique and interesting approaches to the way that they are allocated. Multi-principle allocation systems would be interesting to investigate as future work.

As will be discussed in Chapters 3-5, we not only develop a proposed approach for achieving resource allocation, we also offer a framework that can provide decision support for users interested in learning about the relative value of different possible allocations. We return to reflect on the value of our framework in comparison with related work in Chapter 6.

While we discuss the advantages of our particular proposal for resource allocation for real-time, time-critical dynamic environments in general in Chapter 3, we also return to elaborate on the particular application to healthcare in Chapter 4, and proceed to clarify what we offer for patient scheduling and for handling mass casualty incidents, relative to other approaches, in Chapter 5.

Chapter 3
Multiagent Resource Allocation as Local Search

In this chapter, we outline our proposed approach for multiagent resource allocation using local search and social welfare functions. There are a number of central elements that comprise the calculations; there are also a variety of options for decision makers to view alternative calculations, based on preferences that are specified. We first clearly outline the core process of local search.

Definitions:

- an **individual allocation** is a specification of which resource is assigned to which agent in the system
- a **global allocation** is a pairing of resources to agents for all agents in the system (in both cases, an agent may be allocated the null resource (nothing))
- an **initial assignment** is one that specifies the resources allocated to each agents, to begin the local search for better allocations
- a **swap** between two individual allocations is where the resource assigned to a1 is allocated to a2 and the resource assigned to a2 is allocated to a1
- a **change** for an individual allocation is where the resource assigned to a1 is different from the resource it had been assigned
- a **neighbour** of an existing global allocation is one in which either a swap or a change distinguishes the second allocation from the first

Algorithm:

We decide to adopt a First Improvement Hill Climbing strategy. This means that once we begin with an initial proposed allocation we will select the first neighbor that is better (determined as described below) and we will stop once none of the neighbors are better than our proposed allocation. The search for a better solution requires a metric to determine whether the new proposed solution is an improvement. The metric we use for this determination is a specified social welfare function. In our approach, the initial solution will be determined randomly. A total number of random restarts is specified. This is the number of times the initial allocation is reset (randomly) and the search for a better solution is launched, until termination. The allocation that is best relative to our metric for assessing allocations, from all the attempts through all the random restarts, is output as the Proposed Allocation.

Input: Agents requiring resources, resources that are available, preferred social welfare function

Output: Proposed global allocation of resources to agents

Step 0: Set the total number of random restarts allowable

Step 1: Specify an initial solution (random assignment of resources to agents)

Step 2: Consider a neighboring solution (one identical to the initial solution which has either a swap or a change) and if this offers an improvement, set this to be the Proposed Allocation

Step 3: Continue Step 2 until the stopping criteria is met (current solution has no neighbors that improve)

Step 4: Repeat with a new initial allocation, as many times as specified in the number of random restarts allowed

Step 5: The Proposed Allocation preferred over all the attempted searches for the solution (all random restarts) is output

Further details on how an improvement of one solution over another is modeled are outlined below. As discussed in Chapter 1, we are interested in modeling the cost of a particular resource allocation and in thus preferring an allocation that offers a lower total cost. The total cost of an allocation is calculated as follows.

Step 1: For each individual allocation, determine its cost (based on domain specific features that make this solution less desirable).

Step 2: For the global allocation, determine its cost (based on domain specific global features that make this solution less desirable)

Step 3: Use the specified social welfare function to compute the total costs in Step 1 (considering all individual allocations) and add this to the total cost calculated in Step 2

In the chapter that follows, we clarify possible domain-specific cost functions, for the application of healthcare. The above outlines how what we refer to as the Initial phase of the resource allocation is determined. We are interested in environments that are dynamically changing and would like to support real-time decisions for providing resource allocations. To this end, we allow for what we refer to as an Update phase. Here, parameter values and thus cost function values are changing over time. This then requires a proposal for the frequency with which these values are re-evaluated, what we will refer to as sensing. With changing evaluation of cost functions, then the current allocations can be adjusted further. The process is summarized as below.

Step 1: Set the Sensing Frequency to be X time ticks.

Step 2: At X time ticks after the last check, sense and update parameters.

Step 3: Run the Allocation Algorithm as outlined above.

Chapter 4
Local Search Multiagent Resource Allocation for Healthcare

In this chapter, we outline in detail how our proposed resource allocation framework can be employed for a healthcare application. As discussed in Chapter 1, we want to be able to address two different but related scenarios that can both be characterized as resource allocation for healthcare: allocating patients to ambulances (destined for hospitals) and allocating patients to doctors. We had in mind scenarios such as mass casualty incidents, where an unexpected need for resources to be allocated to a large number of patients may be at play, and where decisions need to be made in real-time, under dynamically changing conditions. We refer to the first stage as the Ambulance stage and the second stage as the Hospital stage.

In order to develop some insight into what should be modeled in this healthcare domain (what properties of the patients, ambulances and hospitals) and from here to specify what should be integrated into appropriate domain-specific cost calculations, we first experimented with what we refer to as the Sufficient Ambulance stage. In this simplified scenario, we omit the consideration of null resources: each patient will receive an ambulance-hospital pair as its assigned resource (where the hospital indicates the destination of the patient).

We also focused entirely on the initial allocation in this simplified scenario. By Initial Allocation we mean a determination of which agents should receive which resources, according to our proposed local search method (with social welfare as its cost metric), where the allocation remains fixed over time as the victims are transported to their hospital destinations. As we will explain further when we discuss the modeling of our full Ambulance and Hospital cases, we allow an Update phase there as well. This means that over time, in the Ambulance stage, the assignment of hospitals to victims may change and in the Hospital stage, the assignment of hospital resources to patients may evolve over time as well, before all the required tasks in the environment are complete.

4.1 Sufficient Ambulance Stage

4.1.1 Example Scenario for the Sufficient Ambulance Stage

We introduce as an example the mass casualty incident of a major car accident on the highway of a city. Note that a preliminary description of our proposal for the Sufficient Ambulance stage appears in [39]. This city has 3 hospitals. The location of the mass casualty incident is in the general vicinity of these 3 hospitals. Each hospital has its own attributes, including distance to location of mass casualty incident, capacity, available resources, and current number of patients. Naturally, if the current number of patients is greater than the capacity then it is over capacity. This situation is not ideal but can happen.

Assuming all the ambulances arrive at the location of the event at the same time, they will fill their ambulance and leave to their assigned hospital. When the victim is put into an ambulance, there is a random chance that the EMS driver will be a skilled veteran or rookie. Also, the victim will have their own attributes besides the EMS driver that they received. Their attributes includes the severity of their injuries, the resources they need at the hospital, and the initial time needed to load them in the ambulance. Some patients will leave the location of the incident faster than others depending on how long it took to find the individual and place them into an ambulance. It is essential to understand that a mass casualty incident could occur anywhere in the city so the hospital with the most available resources may not be the closest, which is unfortunate for the severely injured victims.

For this example (and most of our experimental results for Sufficient Ambulance) the number of people involved in the mass casualty incident is 30 people. It is assumed that there is a one-to-one relationship between the number of injured and available ambulances. Once the ambulance reaches the injured individual, they will assess the severity of the victim's injuries and bring them to the ambulance. The injured individuals may have some broken bones, minor cuts, or be unconscious so to categorize their severity the health conditions from [1] will be used (Section 2.3). Moreover, each victim will require a different amount of resources from the hospital depending on their injury. These resources include medical staff, equipment, and rooms. Each hospital will have its own distance to the mass casualty incident, where it is better to have the hospital be closer to the incident. The resources available in the hospital will not always meet the patient's requirements. This does happen in the real world, which causes delay in the patient receiving treatment. The patient that needs to wait is usually the healthiest. It is assumed that a healthier patient understands that they will likely have a lower priority than the patient with critical health.

When the victim is inside the ambulance, the EMS driver will communicate with the central EMS department. This department will run the proposed local search algorithm on this situation and let the driver know which hospital is their destination. To make this decision, the department will use certain local search strategies and social welfare calculations. It should not be thought of as the EMS drivers competing with each other for the best hospital but rather that they cooperate to find the best result for this situation.

4.1.2 Constraint Satisfaction Perspective

A constraint satisfaction problem is described as decision variables, which are assigned values in the domain, and a set of constraints [38]. The example in this section is an optimization problem – we would like all victims to receive the best possible healthcare, through the allocation of the resources – so the goal is to find a solution that optimizes an evaluation function (in our case this will be a cost function). The decision variables in this problem are the patients. Each one of the patients will be assigned a hospital; this provides a value for each variable. This must be done subject to a number of constraints, outlined below. There are many constraints with this problem but given that this example reflects the real world, returning the no solution is not an option. This is why all the constraints are soft and the cost function will be used to optimize the solution.

There are three soft constraints that influence the cost function. The first soft constraint is the **capacity** of the hospital. If there are more patients assigned to a hospital than its capacity then the cost function should return an unfavorable result. The second soft constraint relates to the **resources**. If the hospital that the patient is assigned to has fewer resources than required then the cost function should return an unfavorable result. This is because this situation represents the patient having to wait at the hospital for the resources to become available. It does not cause the entire problem to fail but rather is an inconvenience to the healthier patients who will need to wait. The final soft constraint relates to the EMS **driver** driving the ambulance to the hospital. The calculation very much depends on the **severity** of the injured patient. One component of the total cost function should therefore be the **cost of the drive**. Preferred cost values are given to better health, closer hospitals, a skilled veteran driver, and less initial time. If the patient's severity is fair and the drive is far then a slightly unfavorable result is given but better EMS driver skill and initial wait time will improve this result. If the patient's severity is serious and the drive is far then a more unfavorable result is given but again the EMS driver skill and initial wait time will improve this result. Finally, if the patient's severity is critical and the drive is not close then the most unfavorable result is given but, like the previous times, the EMS driver skill and initial wait time will improve this result.

This example has been described in such a way that optimization would require optimizing the total cost function. The **total cost** function is the combination of each patient's overall cost function, influenced by the capacities of the hospitals. The approach to combining each value will depend on social welfare function used. The additional components for the total cost function will be explained after we provide more detail on the proposed modeling of the environment.

4.1.3 Modeling

Figure 4.1 summarizes the various concepts that we want to model and their parameters. This is depicted according to our implementation, in Java displayed in further detail Appendix A.

When a victim is paired with a hospital, it is placed in the SingleSolution class. Calculating the cost for this assignment is now determined on the basis of our **CostDrive** function and also a **CostResource** function which will depend on the resources needed by the patient and the resources available at the hospital. If the victim's resources are not met then a penalty is imposed.

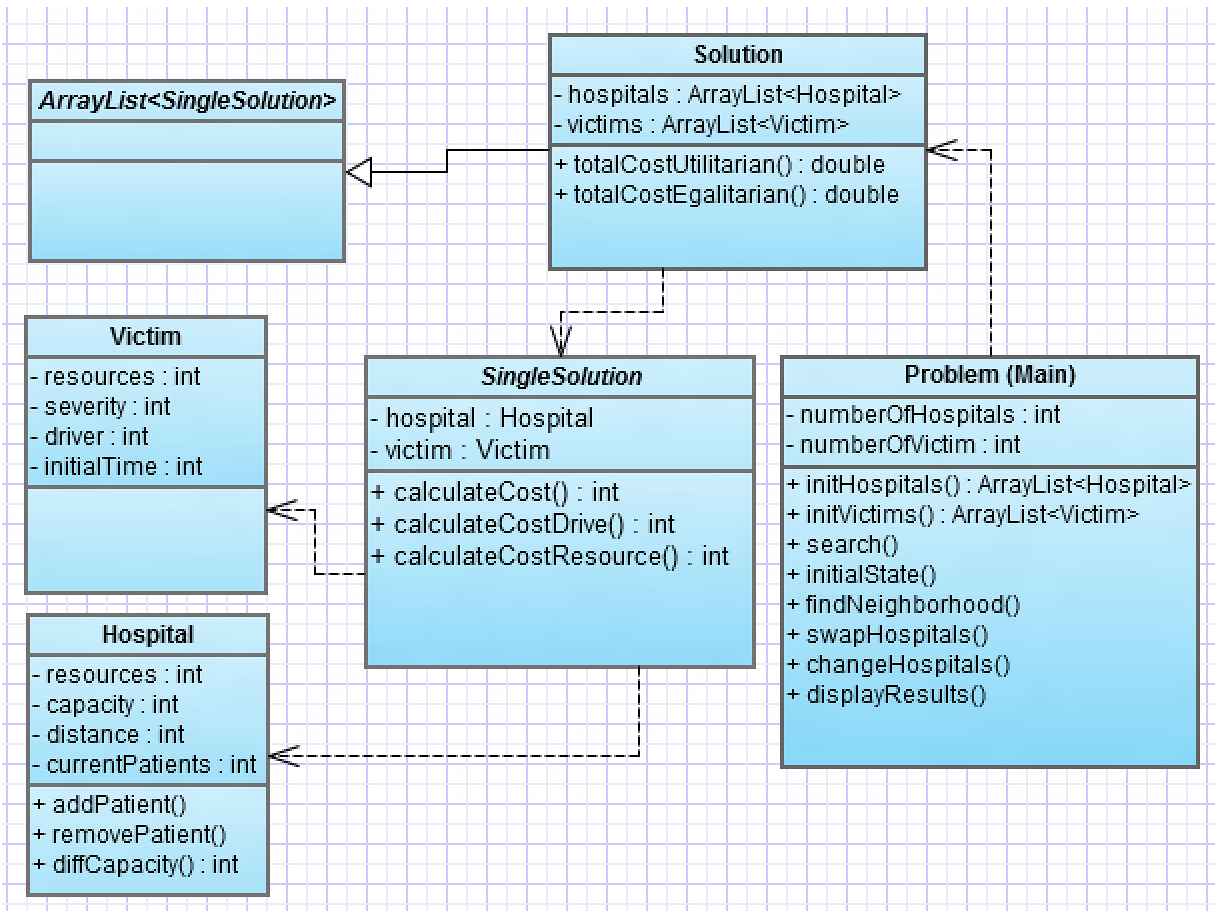

Figure 4.1 – Class diagram of the problem

The Solution class contains the assignments of all the patients to a hospital. It can used to calculate the total cost by combining each individual overall cost. What should also be included at this point as part of the total cost is a capacity penalty producing a less favorable cost if the hospital is overcapacity. The search() algorithm (see Algorithm 4.1) performs a local search to assign patients to a hospital. This encapsulates our proposed approach in full, which embodies a hill climbing method, using a first-improvement approach for adjusting the solution. In our context, a neighboring solution is one which differs by a slight change in our current solution (either swapping hospitals between victims or changing hospitals for a victim). First-improvement will swap values as soon as a better neighbor is found and will continue until no better one is possible, resulting in the final proposed allocation. Note that a better solution is one that has a better total cost.

The initialState() function determines the initial state for the problem. For exploration of Sufficient Ambulance, we considered three approaches for the initial state. The first is **evenly** distributing the victims to the hospitals. The next approach is assigning all the victims to the **same (single)** hospital. The final approach assigned the victims to a **random** hospital. In our validation, we experiment with these different methods for obtaining the initial solution and compare their performance (see Appendix A for details).

The findNeighborhood() function is used to determine the possible neighborhood of the current state. This function uses the stopping criteria of no improvements in the neighborhood and has a **first-improvement strategy** to move to the next neighbor. Swapping hospitals between victims or setting a new hospital for a victim may be introduced in order to improve the total cost. For example, suppose that Victim A is originally set for Hospital X, Victim B is set for Hospital Y, and no one is headed to Hospital Z. A swap between Victim A and Victim B would send Victim A to Hospital Y and Victim B to Hospital X. An example of a change would be for Victim A, which was originally set for Hospital X, to now head to Hospital Z. In the end, we would want to the display various results, including the total cost of the initial state, the final total cost, the number of neighbors found, the number of hospitals that are over capacity, the total number of patients that are over the capacity in each hospital, the number of patients waiting in hospitals for resources, the number of hospital swaps performed, the number of hospital changes performed. The final total cost that would arise if a different social welfare function were applied is also instructive to present.

Algorithm 4.1: Local search to assign patients to hospitals

```
Input: Patients, Hospitals
Output: Solution (assignments of resources to patients)
Initialize solution = initialState();
loop until stopping criteria met do
 for each singleSolution1 in solution
  for each singleSolution2 in solution
   if patient in singleSolution1 != patient in singleSolution2 then
    if swapping hospitals between singleSolution1 and singleSolution2 is better than the current
assignment then
     A swap has occurred with a neighbor solution;
     Go back to the loop;
    else
     Continue in the for loop;
    end
   else
    for each hospital in hospitals
     if changing this hospital to the singleSolution1 is better than the current assignment then
      A change has occurred with a neighbor solution;
      Go back to the loop;
     else
      Continue in the for loop;
     end
    end
   end
  end
 end
end
return solution;
```

4.1.4 Experimental Results

The experimental results were produced by running the local search on 100 different generated scenarios. Appendix A displays these results in full. There were three different approaches to determine the initial state. **Even** distribution, **single** distribution, and **random** restarts (where the problem is attempted (in our case) three times and the best result is returned). In order to perform the local search that yields the proposed resource allocation, a choice for determining the initial state and a choice for the social welfare function must be made. In our experiments we focused on Utilitarian and Egalitarian social welfare. The choice of social welfare affects how the search proceeds, what swaps are made, what reallocations are done, etc. until the final allocation is produced. There are therefore 6 options possible. From our experimental results we determine that using random restarts and a Utilitarian social welfare function is the best.

4.1.5 Summary

This preliminary exploration served to inform our final decisions for the parameters to be modeling and the cost functions to be using. As a result of these experiments we also decided to adopt the strategy of random restarts for setting initial allocations for our search. We also decided to use Utilitarian social welfare predominantly during our subsequent experiments, in order to view representative outcomes of our local search approach in full detail. We also acquired some insight into what elements to include into our cost functions and learned the value of considering alternative options when presenting information to users. Since time to completion for the runs was encouraging, we felt that this kind of system would be of value towards actual deployment.

4.2 Ambulance and Hospital Stages

This template numbers the headings in a legal numbering format. You may choose not to number headings, or to choose a different numbering style, perhaps **A, B, C—1, 2, 3 –a), b), c)** etc. The numbering style can be changed via the **Home** tab in the **Styles** section using the **Multilevel List** button.

4.2.1 Example Scenario for the Ambulance and Hospital Stage

We return again to our example of the mass casualty incident. We now begin to sketch solutions that go beyond what we have referred to as an initial allocation, coping with dynamic environments and required updates. We also remove some of the simplifications of the Sufficient Ambulance stage.

When ambulances arrive at the incident, there may be fewer ambulances than there are victims. Either more ambulances are on the way or the victims will need to wait for the ambulances to return from dropping off a victim. This situation is not ideal for the victims but can happen in the real world. An ambulance can only take one victim at a time, which makes an unshareable resource. Note that victims can still share the hospital resource (sending multiple victims to the same destinations). While at the location of the incident, victims can change between ambulances for an initial period of time but once the ambulance begins to drive the victim will stay in the ambulance until it reaches a hospital.

The victim will be assessed of its injuries by the ambulance staff at the incident in order to determine its level of severity. Moreover, resources needed by the victim are evaluated during this assessment. In order to allocate a victim to an ambulance and hospital, the central EMS department will run local search on the situation. The EMS drivers will be informed of the victims who will receive an ambulance and the hospital that will be their destination.

Since this scenario is dynamic, another set of allocations will be made after a certain period of time (i.e. what we refer to the Update phase). Though a solution may be appropriate for a current situation, the health of an injured individual can change suddenly causing the problem to change and requiring a new solution, which is mentioned in Rossi, van Beek, and Walsh's work [38]. During this time many events can happen such as new ambulances arriving at the incident and a change in the victim's severity. (Future work could address additional considerations, such as a sudden road block while driving to a hospital changing the expected arrival time). This relates to the works of [4], [12], and [42] where restrictions or relaxations are applied to the problem based on the changes that occurred since the previous problem. Information about the scenario needs to be up to date in order to have a best possible allocation. Determining with what frequency to update and reset the allocation will be mentioned as part of our discussion of the decision support. In our discussion of this example scenario we will assume that this will be at every time step.

As the first stage (Ambulance) ends, the second stage (Hospital) picks up right where the other left off and includes a lot of overlap with the first. When the injured individuals reach the emergency room of the hospital, they must be allocated resources. In our work, we use a reuse-previous-solution approach, as mentioned in work by Rossi, van Beek, and Walsh [38]. Once in the hospital, the injured individuals become patients and the multiagent resource allocation definition changes. The agents are all the patients in the hospital with medical staff making their decisions. It is important to note that the attributes being modeled and their current values are transferred from when the patient was in the ambulance. A new attribute for the patient will be the procedure that they require. The procedure will be performed by the resources, which includes the doctors, nurses, other medical staff, equipment, and rooms. Each resource can perform a procedure with a certain skill. A patient is assigned a resource by the central EMS department, which runs a local search on the situation. Since this scenario is dynamic, another set of allocations will be made after a certain amount of time. During this time, patients require different resources, new patients may arrive, and resources can become available. This constitutes the provision for Update at the Hospital stage.

In comparison to the previous Ambulance stage, resources, such as medical staff, are bundled together, may not be shared, and must be allocated to the patient until completion. (During the first few moments of allocation, we do allow a patient and resource to be reallocated). There is also the issue of communication that the previous scenario did not need to address. Communication was reliable in the previous scenario but in this scenario communication between medical staff may not always be possible. For example, during a surgery some of the doctors will not be available for communication. In addition, the room, medical equipment, and medical staff involved in the procedure may not be available for negotiating and changing resources. During this time, when plans are modified then this plan should not change, which relates to Mittal and Falkenhainer's work [29] about active variables. For example, completing a cast on the third floor should not change the plan of surgery taking place in the first floor.

4.2.2 Parameter Modeling

We spent considerable effort in trying to characterize what parameters should be modeled in this particular domain. The set of parameters and their possible ranges of values are provided in Tables 4.1, 4.2, and 4.3 below for the Ambulance stage.

We envisaged using these parameter values as part of our cost function calculations and thus for the core set of attributes in each table, we consistently mapped these to values that were largely on par with each other. We return to discuss possible experimentation with alternate parameter values, in Section 6.3. An example of calculating the total cost functions using these parameters is provided in Appendix B.

Table 4.1 – Victim Attributes (Ambulance Stage)

Attribute	**Possible Values**
Resources	• Low (1) • Medium (2) • High (3)
Severity	• Critical (1) • Serious (2) • Fair (3) • Good (4)
Initial Time	Integer Value • Less than 10 (1) • Between 10 and 15 (2) • Between 15 and 20 (3) • Between 20 and 30 (4)

	• Greater than 30 (5)
Availability	• Available (0) • Unavailable (1)
Lottery	Integer value from 0 to N – 1, where N is the number of victims
Fcfs	Integer value from 0 to N – 1, where N is the number of victims
Sickest first	Integer value from 0 to N – 1, where N is the number of victims
Youngest first	Integer value from 0 to N – 1, where N is the number of victims
Lives saved	Integer value from 0 to N – 1, where N is the number of victims
Prognosis	Integer value from 0 to N – 1, where N is the number of victims
Instrumental	Integer value from 0 to N – 1, where N is the number of victims
Reciprocity	Integer value from 0 to N – 1, where N is the number of victims
Age	Integer value
Survival percentage	Integer value from 0 to 100
Expected years	Integer value
ID	Integer value

Table 4.2 – Ambulance Attributes (Ambulance Stage)

Attribute	**Possible Values**
Driver Type	• Basic (1) • Experienced (2)
Initial Time	Integer Value • Less than 10 (1) • Between 10 and 15 (2) • Between 15 and 20 (3) • Between 20 and 30 (4) • Greater than 30 (5)
Availability	• Available (0) • Unavailable (1)
Longitude	Integer Value
Latitude	Integer Value
Driving	Boolean Value
ID	Integer value

Table 4.3 – Hospital Attributes (Ambulance Stage)

Attribute	**Possible Values**
Resources	• Low (1)

	• Medium (2) • High (3)
Capacity	Integer Value • Low – 8 new victims (1) • Medium – 12 new victims (2) • High – 15 new victims (3)
Current Number of Patients	Integer Value
Arrived Patients	Integer Value
Distance	Integer Value • Close – Less than or equal to 3 (1) • Medium – Less than or equal to 8 but greater than 3 (2) • Far – Greater than 8 (3)
Longitude	Integer Value
Latitude	Integer Value
ID	Integer Value

First of all, the severity of the patients is an important parameter and for this we decided upon a solution that emulates that of [1], distinguishing good, fair, serious, and critical. Note that InitialTime represents time from the beginning of the incident until the victim is left in the ambulance. The Availability attribute describes whether the victim has been given resources (unavailable) or not. Various parameters are also modeled in order to model the patient when considering various principles of allocation (like Youngest First). Age is included for this reason. The Survival Percentage is intended to reflect the likelihood of surviving based on age and severity, with Expected Years representing the number of years a victim is expected to live, after the incident, based on Survival Percentage. Parameters connected with Principles of Allocation (like Sickest First) are intended to represent how this victim stands in the ranking among all patients, when applying that principle. Further discussion of Principles of Allocation appears in Section 4.3.1. Note as well that ambulances are non-shareable resources (one patient per ambulance) whereas Hospitals are shareable. Various parameters help to represent location and whether the ambulance is en route to a hospital or not. In our Implementation (Chapter 5) we use Manhattan distance in order to calculate the distance parameter of the Hospital class.

A few additional notes are important. There are two distinct challenges in developing the characterization outlined here. The first is to include parameters that are good choices for effectively modeling what should be influencing the resource allocation decisions. The second is to determine how to set the possible range of values for each of these parameters. This is an important consideration. The cost functions that drive the decisions about which allocations will be preferred will be adding together costs that derive from different considerations (of the patients, ambulances and hospitals). As can be seen in the details provided in Tables 4.1, 4.2, and 4.3, we made an effort to put differing parameters into similar ranges of possible values. We illustrate the interplay of parameter values for cost calculations later in this thesis (Appendix B).

There was another decision made with respect to parameter values, one that was influenced by our desire to be able to perform quick calculations, to make rapid determinations of possible allocations. Here we mapped various ranges of possible values into a single determining value (for example hospital distances) and ensured finite discrete values. Rossi, Van Beek, and Walsh's [38] work describes how efficiency may be limited in practice if these considerations are not made.

Once we settled on the parameters to be modeled, we then needed to propose specific combinations of considerations as the cost functions that would drive the determination of the preferred allocations. The particular cost functions that we propose are outlined below. While motivated by insights from our Sufficient Ambulance stage, these cost function also go beyond, to introduce new elements. In particular, the cost functions for Ambulance and Hospital stages are distinct.

We begin with a clarification of what we modeled for the Ambulance stage. We then move on to discuss the cost calculations for the Hospital stage. While many of the parameters that are modeled and combined and several of the key concepts that have cost considerations are raised in the calculations of both Ambulance and Hospital stages, there are some additional elements introduced for the Hospital scenario that we will pause to clarify in full detail. Note that these additional considerations cause us to generate some additional kinds of graphs when validating our approach as well, for Hospital vs. Ambulance stages, discussed in greater detail in Chapter 5.

4.2.3 Ambulance Stage

We first clarify what is proposed for the Ambulance stage. The parameters that are modeled and their ranges of values are as in Tables 4.1, 4.2, and 4.3.

Our local search algorithm will settle on the allocation that minimizes the total cost. Algorithm 4.2 below shows the different components of this total cost calculation for the Ambulance stage: i) a cost arising from the individual allocations for each of the victims (SingleSolution costs) aggregated according to the specific social welfare function selected as part of the input ii) a cost arising from the ambulances namely a penalty applied each time a victim receives no ambulance iii) a cost arising from the hospitals namely a penalty for being over or under capacity (with the former being a greater cost).

Algorithm 4.2: totalCostAmbulance()

```
Input: Each assignment of victims to ambulances and hospitals (singleSolution), Hospitals,
Ambulances, Social welfare function
Output: Final cost based on social welfare
if social welfare is Nash Product then
  totalCost = 1;
else
  totalCost = 0;
end
for each singleSolution
  cost = singleSolution.calculateCostAmbulance();
  Combine cost to the totalCost according to the social welfare function;
end
for each ambulance in Ambulances
  cost = 0;
  if null ambulance then
   cost = AMBULANCEPENALTY;
  end
  Add cost to the totalCost;
end
for each hospital in hospitals
  cost = 0;
  if hospital is over capacity then
    cost = number of victims over capacity * HOSPITALOVERCAPACITYPENALTY;
  else
    cost = number of victims under capacity * HOSPITALUNDERCAPACITYPENALTY;
  end
  Add cost to the totalCost;
end
return totalCost;
```

The cost function used within each individual allocation is displayed below:

$$\begin{aligned} CostAmbulance = {} & CostDrive\ (v,a,h) \times\ DriveWeight +\ VicimWait\ \times\ TimeWeight \\ & +\ CostRes\ (v,a,h)\ \times\ ResWeight + PrincipleRankingCost \times\ PrincipleWeight \end{aligned}$$

The central elements of this cost function are i) cost of allocating a resource based on the drive ii) cost of victim waiting for the resource and iii) cost of allocating the resource based on the victim's resource needs. We will explain the PrincipleRankingCost in Section 4.3.1, as we explore some of the options that we make available to decision makers interested in learning about possible allocations.

The three component cost functions are further clarified as below:

The CostDrive function can be seen below where hdist represents the hospital distance from the victim to the hospital, maxVictim represents the best possible severity, vicSev represents the victim's severity level, maxType represents the best possible skill type for an ambulance driver, ambType represents the ambulance driver type, and noAmbulanceDriveCost is a cost for this victim not receiving an ambulance. The reason for these values is to ensure that a victim of severity 1 (the most critical kind of victim) not receiving a resource returns a higher cost than a victim of severity 4 not receiving a resource. For example, suppose there are two victims with severity 1 and severity 2 that are close distance to the hospital and two ambulances of differing skills. There will be a lower cost of assigning the victim with a worse severity to a better skilled ambulance and assigning the victim of severity 2 with the ambulance of lower skill. In addition, if there was only one ambulance then assigning the victim with a severity of 1 to the resource results in a lower cost.

$$CostDrive = \begin{cases} 1 + hdist + \Big((maxVictim - vicSev) \times (maxType - ambType) + \big(maxType \times (vicSev - 1)\big)\Big), & resource\ is\ not\ null \\ noAmbulanceDriveCost\ -\ vicSev, & resource\ is\ null \end{cases}$$

Cost of wait has upper (5) and lower (1) values as indicated in Tables 4.1 and 4.2 (Initial Time). The CostRes function can be seen below where hRes represents the hospital resources, vRes represents the resources that the victim requests, LOWRC represents a low resource cost, MEDRC represents a medium resource cost, and HIGHRC represents a high resource cost. The reasoning for these returned values is to have a low costs for resources being met but at the same time minimizing the difference between the resource requested by the victim and available from the hospital.

$$CostRes = \begin{cases} hRes - vRes + LOWRC, & resource\ at\ hospital\ is\ better\ than\ or\ equal\ to\ victim's\ request \\ vRes - hRes + MEDRC, & resource\ at\ hospital\ is\ worse\ than\ victim's\ request \\ HIGHRC, & resource\ is\ null \end{cases}$$

4.2.4 Hospital Stage

For the Hospital stage several of the parameters and parameter values carry over from the Ambulance stage, such as PatientSeverity. The relevant tables are Tables 4.4 and 4.5.

Table 4.4 – Patient Attributes (Hospital Stage)

Attribute	Possible Values
Procedure	• Procedure A (1) • Procedure B (2) • Procedure C (3) • Procedure D (4) • Procedure E (5) • Procedure F (6)
Severity	• Critical (1) • Serious (2) • Fair (3) • Good (4) • Discharge Patient (5)
Initial Time	Integer Value • Less than 10 (1) • Between 10 and 15 (2) • Between 15 and 20 (3) • Between 20 and 30 (4) • Greater than 30 (5)
Availability	• Available (0) • Unavailable (1)
Lottery	Integer value from 0 to N – 1, where N is the number of patients
Fcfs	Integer value from 0 to N – 1, where N is the number of patients
Sickest first	Integer value from 0 to N – 1, where N is the number of patients
Youngest first	Integer value from 0 to N – 1, where N is the number of patients
Lives saved	Integer value from 0 to N – 1, where N is the number of patients
Prognosis	Integer value from 0 to N – 1, where N is the number of patients
Instrumental	Integer value from 0 to N – 1, where N is the number of patients
Reciprocity	Integer value from 0 to N – 1, where N is the number of patients
Age	Integer value
Survival percentage	Integer value from 0 to 100
Expected years	Integer value
ID	Integer value

Table 4.5 – Resource Attributes (Hospital Stage)

Attribute	Possible Values
Type	• Type A (1) • Type B (2)

	• Type C (3) • Type D (4) • Type E (5) • Type F (6)
Capability	• Capability A (1) • Capability B (2) • Capability C (3) • Capability D (4) • Capability E (5)
Skill	• Low (1) • Medium (2) • High (3)
Initial Time	Integer Value • Less than 10 (1) • Between 10 and 15 (2) • Between 15 and 20 (3) • Between 20 and 30 (4) • Greater than 30 (5)
Availability	• Available (0) • Unavailable (1)
ID	Integer value

We paused to reflect on what would be best to model, once the victim arrives at the hospital, in need of resources. We first of all decided what procedure a patient needs. We then decided that a patient should be able to acquire a bundle of possible resources (e.g. doctor, nurse, equipment, room). We also considered this to be a non-shareable resource. As another important consideration, we thought about how best to model the scenario as time progresses. As mentioned, we wanted to allow for Update allocations.

In the Ambulance scenario, what we were allowing was to have an ambulance en route to one hospital be sent along to a different hospital instead, en route (based on the proposed allocation of (ambulance, hospital) per victim the value of which is updated at each time step, as sensing introduces new parameter values). Note that we defer the discussion of sensing to Section 4.3, as part of our clarification of Decision Support.

In modeling hospital resources, therefore, we decided we need to model a new entity, Resources. This is non-shareable as it represents the medical staff members and equipment that can only be allocated to one patient at a time. Per Table 4.5, the attributes in this class are integers that are used to describe the resource. Each resource has a type that is used to describe the medical staff members and equipment. There are six types with differing capabilities of performing the five procedures needed by the patients. These procedures are performed with a certain skill. Skills are represented by integers as high (3), medium (2), or low (1). A higher skill is desired as a resource will only be met if the skill is medium or high. As in the real world a procedure can be performed with less skill but it is not desired. The skill of a resource performing a procedure can be seen in the table below (Table 4.6). Type A (1), Type B (2), and Type C (3) represents a basic room with a certain set of doctors and nurses. Type D (4), Type E (5), and Type F (6) represents an advanced room with a certain set of doctors and nurses. The difference between a basic and advanced room is that an advanced room will have better medical equipment. Although it may seem that a more advanced room leads to a better performed procedure, this is not always the case. For example, a patient requiring Procedure D to be performed may think that resource Type F is the best choice but would be better off taking resource Type C since it has a better skill. Similarly to the Patient class there are initialTime and availability. The initialTime attribute represents the idle time for the resource and having the same time categories as the Patient class. It is better for the resource to not be idling for a long period of time. The availability attribute represents whether this resource has been allocated to a patient.

Table 4.6 – Skill of resource performing procedure

Procedure / **Resource**	Procedure A	Procedure B	Procedure C	Procedure D	Procedure E
Type A	Low	Low	Low	Medium	Medium
Type B	Medium	Medium	Low	Low	Low
Type C	Medium	High	Medium	High	Low
Type D	Medium	Medium	Low	Low	Medium
Type E	High	Medium	Medium	High	Medium
Type F	Medium	High	High	Medium	High

As mentioned above, the cost calculations for this stage continue to model similar considerations as those in the Ambulance stage and also introduce a few new elements, as clarified below. The total cost function is now as shown in Algorithm 4.3. There are no longer additional costs due to Ambulances or Hospitals but instead due to Resources. Any victims unable to secure a resource will generate an additional burden on the total cost.

Algorithm 4.3: totalCostHospital()

```
Input: Each assignment of patients to resources (singleSolution), Resources, Social welfare function
Output: Final cost based on social welfare
if social welfare is Nash Product then
  totalCost = 1;
else
  totalCost = 0;
end
for each singleSolution
  cost = singleSolution.calculateCostHospital();
  Combine cost to the totalCost according to the social welfare function;
end
for each resource in Resources
  cost = 0;
  if null resource then
   cost = RESOURCEPENALTY;
  end
  Add cost to the totalCost;
end
return totalCost;
```

The SingleSolution cost for this stage is computed according to the formula below:

$$\begin{aligned} CostHospital = {} & CostPatRes\ (p,r) \times PRWeight + PatientWait \times TimeWeight \\ & + CostPatResSkill\ (p,r) \times SkillWeight + PrincipleRankingCost \\ & \times PrincipleWeight \end{aligned}$$

As before, we defer the discussion of PrincipleRankingCost to Section 4.3. There is a resource cost once more (CostPatRes) but in this case a resource assigned is more costly if a patient with high severity does not receive resources. The CostPatResSkill component allows a greater cost if the resource assigned to the patient is at a lower skill level and the PatientWait component introduces larger costs when patients have had to wait longer before securing a resource.

The functions above are explained in detail as follows:

The CostPatRes function can be seen below where maxPatient represents the best possible severity, patSev represents the patient's severity level, maxType represents the best possible resource type, resType represents the resource type, and noResourceCost is a cost for this patient not receiving a resource. The reason for these values is to ensure that a victim of severity 1 not receiving a resource returns a higher cost than a victim of severity 4 not receiving a resource. For example, there are two patients with severity 1 and severity 2 and two resources of differing types. There will be a lower cost of assigning the patient with a worse severity with a better type resource and assigning the patient of severity 2 with the worse resource. In addition, if there was only one resource then assigning the patient with a severity of 1 to the resource will result in a lower cost.

$$CostPatRes = \begin{cases} 1 + ((maxPatient - patSev) \times (maxResourceType - resType) + (maxResourceType \times (patSev - 1)), & resource\ is\ not\ null \\ noResourceCost\ -\ vicSev, & resource\ is\ null \end{cases}$$

Cost of wait has upper (5) and lower (1) values as indicated in Tables 4.4 and 4.5. The CostPatResSkill function can be seen below where skillOfResource represents the skill of the resource (if resource is null then the value of -1) and a constant PRSKILLCOST. The reasoning for these returned values is to return a better value (lower number) for the resource performing a better skilled procedure on the patient.

$$CostPatResSkill = PRSKILLCOST - skillOfResource$$

4.3 Decision Support

We are interested in developing a corrector system in a health care environment. The proposed system acts as a decision support system that will help medical staff members make appropriate decisions for their patients. If their decision will lead to a suboptimal result then the system will alert the user about the incorrect decision and recommend a decision that will lead to an optimal result. Of course, if the user’s decision is optimal then the system will just confirm that they should continue in the direction that they are going. It will be the responsibility of the medical staff to make the final decision given the recommendation of the system.

A design approach was used to develop this system with the main intention of use by an emergency department. The individuals intended to use the system are first of all doctors, emergency medical services (EMS) drivers, and any other medical staff member. Given the nature of ambulances and hospitals, the system should be able to handle the dynamic environment. This means that the system should keep track of the changing environment since the current solution may only be appropriate for the current situation.

The goal of the system is to recommend optimal allocations of resources to the patients. These resources can be ambulances, hospitals, medical staff, or medical equipment. In the real world, resources may be limited and can sometimes be unavailable. Using a local search, the system tries to find an optimal solution to the situation. The allocation value will be used to evaluate the solution. Social welfare and principles of allocation are an important factor for the policy.

Decision making in the emergency department are affected by numerous factors. These decisions are made in a stressful situation and during a short period of time. The better the decision made, the better the result is for the patient. Motivation for designing this system is to help medical staff members in their decision, which will allow patients to recover as well as possible. Medical professionals would most likely be specifying their preferred allocation to the system running in decision support mode, to see whether these are optimal.

Another set of users that are important to support are administrative individuals in the emergency department. These users would be interested in determining the best policy to implement into the system by analyzing different scenarios in order to conclude the best fit. Therefore they would be most likely to specify their preferred social welfare function to the system in order to determine whether they want to remain with that choice.

Our algorithms for resource allocation are designed to output proposed allocations of resources. But we envision also operating in a kind of Decision Mode, whereby users can request to view particular information, especially to learn about how competing choices would compare, if chosen to drive the resource allocation algorithm.

In the subsections that follow, we clarify some of the choices that we allow our decision makers to specify, in order to view the resulting resource allocation or to compare the outcome from two differing specifications. In Chapter 5, we devote particular effort to providing a valuable toolkit for decision makers to view the output from competing preferences for the input parameters.

4.3.1 The Concept of a Policy

Our design for MARA using local search and social welfare functions is intended to provide for decision makers some latitude to view alternate solutions and the allocation values they will generate (the costs, which should be minimized). Towards this end, we have provided as part of our design what we refer to as the specification of a policy: a combination of a preferred social welfare function and a preferred principle of allocation, to be respected within the allocation algorithm.

The social welfare functions that we experimented with as part of our validation included: Utilitarian, Egalitarian and Nash product. The principles of allocation available to explore were selected from the list provided in [34] and included: lottery, first come first served (FCFS), sickest first, youngest first, saving the most lives (we refer to as lives saved), prognosis, instrumental value (we refer to as instrumental), and reciprocity.

We note here that our framework is general enough to support differing application areas. While in this section we describe what we supported for ambulance and hospital scenarios for the healthcare domain, there are intuitive counterparts that could be introduced for other application areas. For example, if trying to specify a preferred principle for forest fighting, one could select options such as most-injured (instead of sickest first) and saving the most trees (lives saved). We discuss this path for future research further in Section 6.3.

The costs that arise with a particular allocation will differ, depending on the social welfare function that is specified. This arises, for example, from the loop within (Algorithm 4.2) where the costs from each SingleSolution for a particular victim are combined based on what the social welfare function dictates.

There are two distinct uses examining principles of allocation, when determining possible allocations. The first is to consider this as an alternative method for deciding the final allocation: one that will not be based on local search but that will, instead, be developed by examining each patient, in turn, deciding which resource should be assigned to that patient based on its ranking from the specified principle. The second is as part of the decision maker's specified preferences e.g. Utilitarian social welfare but also Youngest First.

4.3.1.1 Clarifying Example: Local Search vs. Principle of Allocation

An example will serve to explain the first usage: what kind of resource allocation would be proposed, if the algorithm were based entirely on a particular principle of allocation.

As an overview, the principle approach takes a look at the individual that is being allocated a resource and does not appreciate that if this individual takes a slightly worse resource for them then another individual can end up with a better resource in such a way that the collection of agents will improve the global utility. In our local search based approach we begin with an initial suggestion for an allocation and continuously change the allocation to improve the global utility (i.e. minimizing the total cost). This important distinction between these two options for MARA will be apparent in the discussion that follows.

Below we clarify an appropriate allocation of resources for our Ambulance stage. In this small example, there are 4 victims: v1, v2, v3 and v4. Their order in terms of severity (most severe to least severe) is v3, v2, v1 and v4. Their order of arrival (first to last arrival) however is v2, v4, v1, v3.

Suppose there are two ambulances, a1 and a2 and three hospitals h1, h2 and h3. We will populate these resources with certain parameter values from Appendix B as follows: a1 has a skilled driver, a2 has a skilled driver, a3 has a less skilled driver; h1 is close with high resources, h2 is close with low resources, h3 is far with low resources.

To decide an allocation based simply on principle, the allocations that are proposed in these cases turn out to be (v1, a3, h3), (v2, a2, h2), (v3, a1, h1), (v4, null, null) for sickest first and (v1, a3, h3), (v2, a1, h1), (v3, null, null), (v4, a2, h2) for first come first served. It is important to note that no local search is being done here. There is therefore no centralized reasoning about what is best for the society. Each victim, in turn is allocated the resource that is best, based on their ranking according to the principle of allocation (where "best" is determined through the cost functions).

We now continue the example in order to make clear the difference between this kind of algorithm for allocating resources and one that would rely instead on local search. Consider our same 4 victims. Running our algorithm in this case would follow the following steps:

Step 0: decide how many random restarts are allowed (say 3).

Step 1: initialize a solution, from a random selection

A possible allocation would be as below:

(v1, a2, h2), (v2, null, null), (v3, a1, h1), (v4, a3, h3)

Step 2: examine possible neighbors to cycle towards the proposed global allocation

We sketch some hypothetical neighbors here, for illustration:

(v1, a2, h2), (v2, a1, h1), (v3, null, null), (v4, a3, h3)

(v1, null, null), (v2, a2, h2), (v3, a1, h1), (v4, a3, h3)

(v1, a1, h2), (v2, null, null), (v3, a2, h1), (v4, a3, h3)

The best neighbor would be (v1, null, null), (v2, a2, h2), (v3, a1, h1), (v4, a3, h3) (since v1's health is less severe than that of v2 or v3) making it the new current solution. Examining the neighbors of the current solution would determine that (v1, a3, h3), (v2, a2, h2), (v3, a1, h1), (v4, null, null) will be the new current solution. Another cycle of examining the neighbors of the current solution would determine that (v1, a2, h3), (v2, a3, h2), (v3, a1, h1), (v4, null, null) will be the new current solution. After examining the neighbors of the current solution we determine that this current solution is the best. We notice that this solution is a neighbor of the sickest first solution as v1 and v2 swapping ambulances. The idea is that v2 will worsen its cost by taking a less skilled ambulance driver but the cost improvement from v1 receiving a better ambulance driver will cause a better total cost for the global allocation.

In this case, a solution that looks good for the entire society at once is being proposed (in comparison to the principles of allocation scenario that allowed each victim to claim the best resource, in turn).

While the medical professionals who are using our proposed system are allowed to specify a social welfare function, they may want to also respect a kind of principle of allocation within the allocation algorithm. Our framework allows for this. For example, Utilitarian Youngest First would ensure that when the total cost of a set of SingleSolution allocations is determined, it will be calculated on the basis of the Utilitarian welfare function. But when the cost of the individual allocation for a particular patient is examined, allocations which do not offer better resources to younger patients will incur an additional penalty. This is handled by expanding our proposed cost function (Section 4.2.3 and 4.2.4) to also integrate a principle component, as follows: during the local search the PrincipleRankingCost will be a value other than 0. For each victim, the PrincipleRankingCost will be their ranking according to the principle. For example, being the third sickest would add a penalty 2 if using the sickest first principle. This will allow the search to explore different neighbors (due to the different cost) than if PrincipleRankingCost is 0 (which it is with the no principle policy).

4.3.1.2 Clarifying Example: Policy Calculations

What we now refer to as a policy approach is a duo of social welfare function and principle of allocation, with the allowance for no principle to be specified at all, where the additional cost (the PrincipleCostPenalty) of being at odds with the preference of the principle will simply not occur, as part of the calculation for an individual allocation.

At this point a full example of the social welfare function, principle of allocation duo in action would be instructive.

Table 4.7 – Assignments and Assignment Values Based on Sickest First

Assignments Based on Sickest First				
Assignments and Assignment Values	Principle	Egalitarian	Utilitarian	Nash Product
Patient X-Resource A = 2 Patient Y-Resource B = 2 Patient Z-Resource C = 2	No Principle	2	6	8
Patient X-Resource A = 2 + 0 = 2 Patient Y-Resource B = 2 + 1 = 3 Patient Z-Resource C = 2 + 2 = 4	Sickest First	4	9	24
Patient X-Resource A = 2 + 2 = 4 Patient Y-Resource B = 2 + 1 = 3 Patient Z-Resource C = 2 + 0 = 2	FCFS	4	9	24

Table 4.8 – Assignments and Assignment Values Based on First Come First Served

Assignments Based on First Come First Served				
Assignments and Assignment Values	Principle	Egalitarian	Utilitarian	Nash Product
Patient X-Resource C = 3 Patient Y-Resource B = 2 Patient Z-Resource A = 1	No Principle	3	6	6
Patient X-Resource C = 3 + 0 = 3 Patient Y-Resource B = 2 + 1 = 3 Patient Z-Resource A = 1 + 2 = 3	Sickest First	3	9	27
Patient X-Resource C = 3 + 2 = 5 Patient Y-Resource B = 2 + 1 = 3 Patient Z-Resource A = 1 + 0 = 1	FCFS	5	9	15

In this simplified example, there are 3 patients and 3 resources. To simplify the example we assume that Resource A is best for all patients, Resource B is second best for all patients, and Resource C is third best for all patients. Patient X is the sickest, arrived third, and requires Resource A. Patient Y is the second sickest, arrived second, and requires Resource B so if they receive Resource A then it would be better. Patient Z is the third sickest, arrived first, and requires Resource

C so if they receive Resource A or Resource B then it would be better. For this simplified example, we will first assume that if no principle is specified the allocation is the same regardless of social welfare function.

We are interested in clarifying what happens if the user also has a preferred principle of allocation. For example, if the user prefers sickest first then they would expect the following allocation: (X, A), (Y, B), (Z, C).

Earlier, we outlined two individual allocation cost functions, CostAmbulance and CostHospital. For the purposes of this example we will consider a generic cost function for individual allocations (what we refer to as the assignment value), one that first of all assigns an initial value based on whether the patient resource pairing matches the expected pairing. It simply adds in 2 if the resource assigned is the resource they required, adds in 1 if the resource assigned is better than the resource that they required, and adds in 3 if the resource assigned is worse than the resource that they required. We also add in a principle cost penalty discussed below. Note that this simplification ignores many of the attributes and factors in the situation. Likewise the totalCost function used in this example will also be a generic one, one that combines, over all patients the individual allocation values according to a social welfare function (what we call the evaluation function) but that does not include extra ambulance, hospital, or resource costs. Since there are as many resources as patients there are no additional ambulance and hospital costs to consider (per Algorithm 4.2). The total cost function generates what we call the allocation value.

Above we include two tables based on allocation by sickest first and allocation by first come first serve. Each table includes the assignment values and the allocation values. The allocation values specify whether a principle is used and the social welfare function used in the evaluation function. In each table the left half shows assignments and assignment values. The second and third sections clarify how the principle cost penalty adjusts the calculation (adding an extra cost according to the rank of the patient per the principle). Consider for example, using the principle sickest first for Patient X will add 0 to their assignment value, Patient Y will add 1 to their assignment value, and Patient Z will add 2 to their assignment value. When no principle is used then the assignment value does not change.

The right side of each table shows comparison values that a user may want displayed. If, for example, a user is interested in sickest first principle the Table 4.7 would be displayed. The user could specify their preferred social welfare function, for example Utilitarian, and could then see how different specified principles, for example FCFS, would result in differing total allocation values. This may help determine the ideal principle of allocation. In order for a user to decide which principle they might want to adopt, they can look at two tables simultaneously. This is purpose of the no principle choice in the Principle column. For example, if someone who prefers Egalitarian social welfare examining Table 4.7 and Table 4.8 would see that sickest first is a better choice of principle. Note that the values in the right side of the tables are calculated by applying the social welfare function combination rule to the values indicated on the left side of the table.

It is important to note that this was a simplified and somewhat artificial example. When users specify a preferred policy (social welfare function and principle of allocation) the selected principle of allocation will influence the principle cost penalty and thus the single solution cost, while the social welfare function will influence the combination of single solution values within the total cost calculations. As a result the assignment values and allocation values will tend to be different with each differing policy choice.

4.3.2 Sensing and Time Buffers

In this subsection, we clarify two additional specifications that decision makers can provide (which also enable further graphs to be generated that inform decision makers about the influence of setting these parameters differently).

We are operating in dynamically changing environments. As a result, we will need to reassess the value of some of the parameter values, with a certain periodicity. We generalize this consideration in the discussion below as deciding how many "time steps" to wait before a sensing action takes place, which then allows for updated parameter values to dictate the selection of the proposed allocation, when running our local search algorithm.

The notion of a Time Buffer is introduced into our modeling of the domain, to allow a period of time to elapse before the initial allocation of victims to (ambulances, hospitals) (in the Ambulance stage) or patients to resource bundles (in the Hospital stage) is set. For the Ambulance stage, the buffer allows for time before an ambulance leaves the site of the incident to exchange ambulances and victims. This will be beneficial if a new victim is found at the incident or if a victim that did not receive a resource becomes more severely injured. For the Hospital stage, the buffer allow for an exchange of patients and resources before a procedure has gone beyond a critical point in the procedure. For example, resources or patients can be exchanged during the preparation before a surgery but it would not make sense to perform a swap once the surgery has begun.

4.4 Corrector System

While we have provided a full description of our proposed decision support, in this section we reflect on the usage of a system designed to follow this solution, by medical professionals and administration, we term this the corrector system. Note that our solution provides good results for the running time, which is important for the corrector system; this is discussed in Chapter 5.

4.4.1 Scenarios

Before describing the corrector system, we will identify real world situations that would benefit from the system. Suppose that a car accident on a major highway has resulted in five injured individuals. Each individual has their own injury severity and their own requirements to be treated. EMS drivers arrive at the scene of the incident and begin to assess the injured individual. After examining the situation, they make a decision about which hospital the ambulance should set as their destination. Each hospital has its own distance from the location of the accident. Moreover, the hospitals have unique attributes for their resources. Once in the ambulance, the EMS driver will identify the situation to the system and state their decision for the hospital as well as their intended route. The system will evaluate the situation and recommend a better choice if it exists. The EMS driver can choose to accept the recommendation or ignore it. Either way, the EMS driver will clarify the decision to the system. This solution provided by the system will be influenced by the policy of the system. The policy takes into consideration social welfare and principles of allocation. During the ambulance ride to the hospital, the system can be beneficial in the situation where the severity of a patient worsens and the currently assigned hospital is no longer appropriate. Even after the patient arrives at the hospital, the system will still be useful. The corrector system will provide recommendations whether the patient arrives in the ambulance, through their own means, or are already admitted in the hospital. Patients require a medical resource (e.g. medical staff and equipment) in a timely matter for their treatment. The system will alert the appropriately qualified available medical staff to treat the patient and will mention the resources available. The system will alert the medical staff member if they are about to make a suboptimal decision during treatment. An example of a suboptimal decision is the medical staff allocating a resource that would have been better allocated to another patient. Again, the system will recommend a better choice, if it exists, that the medical staff member can choose to accept or ignore.

4.4.2 The System

The corrector system will model each situation as a constraint satisfaction problem (CSP). The search continues until it reaches a stopping criterion. Since medical staff is required to make treatment decisions quickly, the stopping criteria will probably be predetermined based on time. It is important to note the dynamic nature of the health care environment. To handle this property it can beneficial to set the initial assignment in the local search to be the previous solution. Moreover, the system will need to be able to monitor all the patients and resources.

Similar to some of the surveyed literature [13, 21, 31, 33, 40, 43], the corrector system will be a multi-agent system. Patient agents are allocated active resource agents (e.g. medical staff) and passive resource agents (e.g. medical equipment). Monitoring these could be achieved using RFID. Patients will be given their RFID once they reach the ambulance and hospital. The information stored will contain their medical records including their current health and treatment requirements. Resources will have their RFID while they are working. The information stored about them includes their availability, capability, and skill level. WLAN or GPS (Global Positioning System) can monitor the ambulances away from the hospital.

Some techniques from telemedicine [25] could be used to help monitoring. Video and audio communication placed in the hospital will provide a clearer availability than inferring with the RFID. A centralized approach is applied to the system for communication. We refer to the individuals who monitor the communication links and update the system accordingly as medical watchers. RFID will also update the system but can delegate to the medical watchers to ensure accuracy. For ambulance drivers, direct communication takes place between the drivers and the medical watcher. Ambulance drivers will tell the medical watchers information about their patient while assessing and loading the patient into the ambulance. The medical watcher will input the information to the system and provide the result to the ambulance driver. This allows the EMS workers to focus on the patient rather than dealing with the input. It will be the responsibility of the medical watchers to update the system with the final decision made by the medical staff member.

Updating the system happens at fixed time intervals and when events occur. Events happen for many reasons from the patients to the resources. The first reason relates to the patient arriving and leaving the hospital or ambulance. Moreover, a patient's health changing is another event. An event is considered when the medical staff member differs from the allocation specified by the system. Expected events do occur to medical staff members when their shift begins or ends.

The final kind of event is for medical equipment. Medical equipment could be removed, broken or retired, as well as added, fixed or new. Seeing the dynamic nature of this system shows the importance of focusing on time buffers when allocating the resources. Allocations use the time buffers to focus on the near future as well. For example, an ambulance driver is taking a patient to a hospital, where the resources will be reserved for that patient. Those resources will be available for the patient upon their arrival. In addition, the system will only include available resources when determining a solution. For example, a surgeon and the resources in the middle of performing a five hour surgery should not be included when determining the allocation since they are unavailable.

4.4.3 Discussion

The corrector system has its benefits and limitations. Helping the medical staff make decisions about allocating resources to patients is one of the main purposes of the system. The system can consider many different possible allocations and quickly compare them according to the policy. A disadvantage to the system, which is inherited by emergency situations, is that time becomes a factor. The system may need to return the best result seen so far instead of the optimal result because patient is in critical health; this issue is explained further in Section 6.3. An important consideration about the time needed for the system is that the system provides decision support, which means the medical professional already has their course of action in mind and has the final say in the decision. An obvious limitation in the system is the lack of implementation based on medical staff recommendations. Discussing with medical professionals provides a better insight to the requirements of the system since subjectivity is significant in many areas of the system.

The policy, principle of allocation and social welfare, in the system can greatly change the recommendation made. Moreover, the patient's utility function and modeling the situation as a CSP can affect the system. It is important to identify which attributes about the patient are more relevant and how much these attributes affect the patient. While we have devoted considerable – all not to selecting an appropriate set of parameters to model, we realize that further investigations are possible as future work. Implementing a prototype and experimenting with actual medical data would be instructive. Experimentation could cover other areas of local search, such as other algorithms and heuristics, or allocation through other methods, such as MDPs and machine learning.

The system would initially assume that the data given about the patients and resources is accurate. However, a patient's health may be incorrectly assessed. Even further work can involve investigating how to handle noise error. Our work did not fully investigate the way to alert the medical staff of their suboptimal decision. A simple approach would be through their smartphone. The sensors used to monitor the patients and resources can be investigated further in future work. The work of Liao, Fox, and Kautz [24] could be helpful for monitoring since it discusses how learning patterns from GPS traces can be used to determine a person's activities and location. In the corrector system, this technology can be used to better determine a medical staff member's availability. Cryptography, as covered in Li et al.'s work [23], is another part of sensor data that should be considered for the corrector system.

Another area of future work would be considering how the system would change the policy based on the user's final decision. This leads to the question of whether the system should learn from the user or should the user learn from the system. Furthermore, there is the question of whether should the system be fully automated or should a human always be part of the system. Our stance is that a human should always have the final decision.

Chapter 5

Implementation Output for the Healthcare Domain

In this chapter we show output from an implementation of our proposed local search multiagent resource allocation framework, for the healthcare domain. The parameters and parameter values discussed in Chapter 4 are at play, as well as the domain-specific cost functions that were described.

For our validation we randomly generate 100 runs, where the patients' and the resources' attributes are determined randomly for each run. This includes variations to attributes such as severity. For the Ambulance stage, we simulated 30 victims, 20 ambulances, and 3 hospitals. For the Hospital stage, we simulated 10 patients and 6 resources.

For the graphs that follow, we need to select values for all the penalties that exist within our proposed cost function calculations. These values are set below. For Ambulance stage: LOWRC (2), MEDRC (4), HIGHRC (10), AMBULANCEPENALTY (2), HOSPITALOVERCAPACITY (10), HOSPITALUNDERCAPACITY (1), and noAmbulanceDriveCost (18). For Hospital stage: PRSKILLCOST (5), RESOURCEPENALTY (2), and noResourceCost (29). We also set all of the weights in the CostAmbulance and CostHospital function to be 1.

In this chapter, we in fact present a series of graphs. One use of graphs is to provide output which assists decision makers in specifying the required input values. We are offering a valuable toolkit and have devoted effort to considering what kinds of graphs may be useful to display. Decision makers need to specify: a) preferred social welfare function b) preferred principle of allocation (either as part of their specified policy or to see the performance of a MARA approach based alone principle of allocation) (note: that no principle is an option) c) interval for sensing d) length of Buffer Time. We begin with a series of graphs that showcase the Ambulance stage. We first discuss graphs which provide feedback on the relative differences between different social welfare functions and for each social welfare function, the differences between distinct principles of allocation. We also present graphs which assist the decision maker in setting the time interval for sensing and the length of the Buffer Time (the period before the initial allocation is locked in).

From here, we move on to a series of graphs to showcase the Hospital stage. Here we also integrate some graphs to demonstrate our proposed algorithm on a per patient basis as part of our presentation of the output for MARA at the hospital.

After this discussion, in order to summarize, we will reflect on what these graphs, together, provide for decision makers. We will also briefly discuss the inherent value of our proposed local search approach for MARA (in comparison with approaches that rely solely on principles of allocation).

We showcase in this chapter some of the central output that we produced, during implementation. (We use the label VA for graphs validating the Ambulance stage and VH for graphs validating the Hospital stage).Additional graphs are included in Appendix G.

5.1 Random Restarts Decisions

Before we begin, we introduce a series of graphs that examine alternative values for the number of random restarts to be used in our local search approach, done for the Ambulance stage. This discussion is grounded in the context of a Utilitarian social welfare function (graphs for other social welfare functions are displayed in our Extra Validation Appendix, Appendix G).

In the first set of graphs (Figure VA-1 (a), (b), (c), and (d)) we explore the effect of different numbers of random restarts, illustrated for the case where the social welfare functions is utilitarian. In these figures the y-axis represents the allocation value and the x-axis represents the approach taken. Note that there are slightly different ranges on each y-axis. All of the policy values are better than those for principles of allocation. It can be seen that as the number of random restarts increases, the results are more favorable. Moreover, all the policy approaches seem to be very even. These graphs might help a system designer how many random restarts to use. The accompanying table (Table VA-1) displays the average running time (computed over the 100 runs) for each MARA approach. Looking at the policy based choices all of these approaches take less than 1 second. This shows that having 100 random restarts is still a viable approach since the result can be determined quickly. For the remainder of this section we will consistently set the number of random restarts used in our policy based approach to 100.

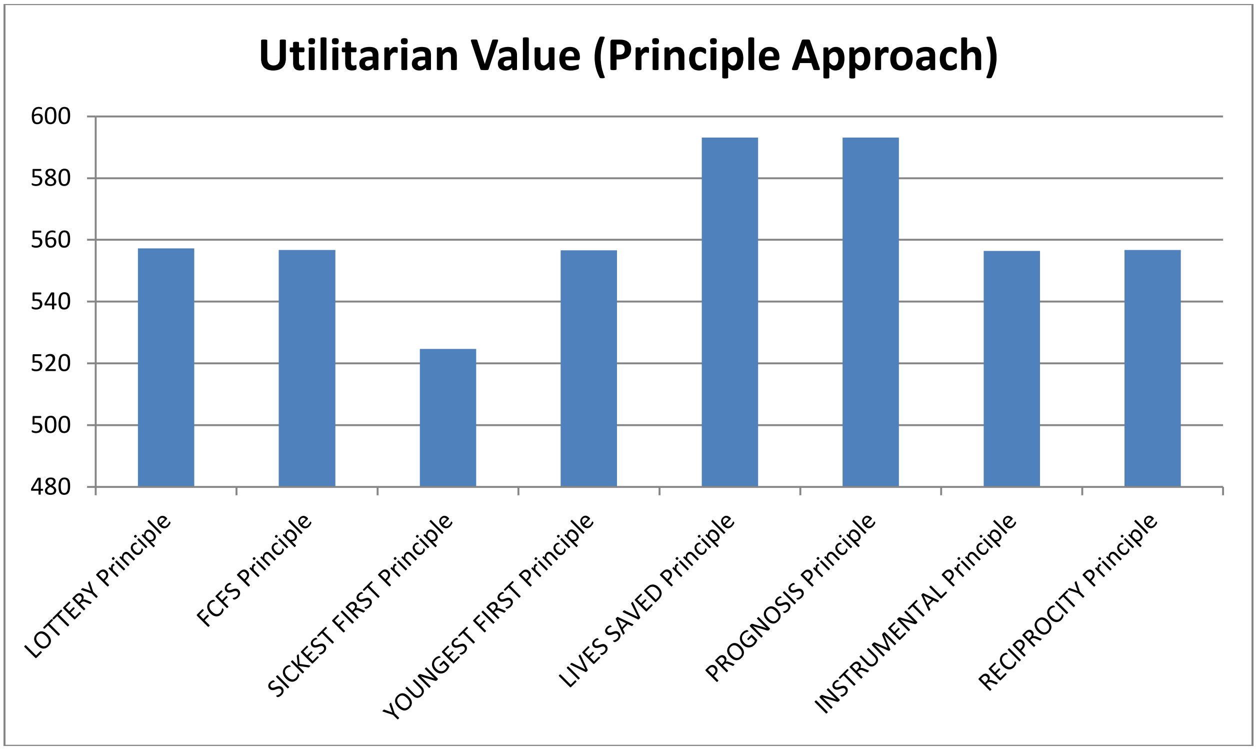

Figure VA-1a – Ambulance: Initial Utilitarian Allocation Value Principle Approach

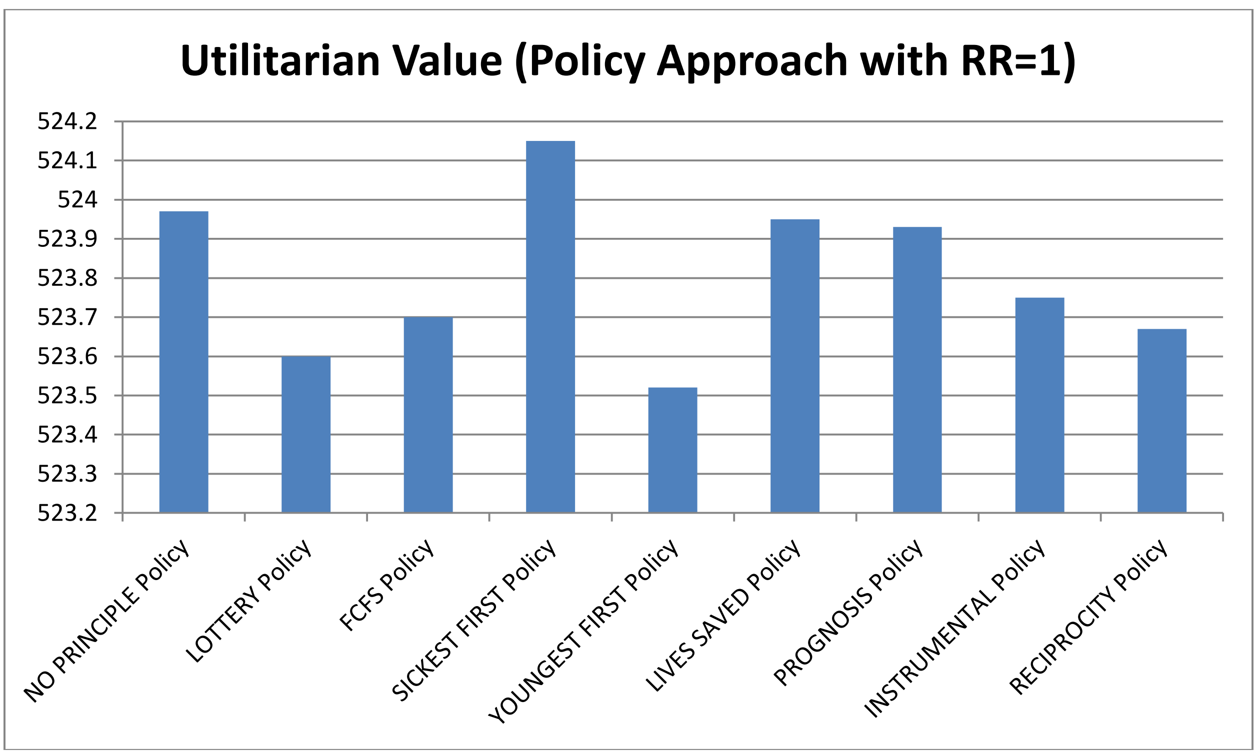

Figure VA-1b – Ambulance: Initial Utilitarian Allocation Value Policy Approach with RR=1

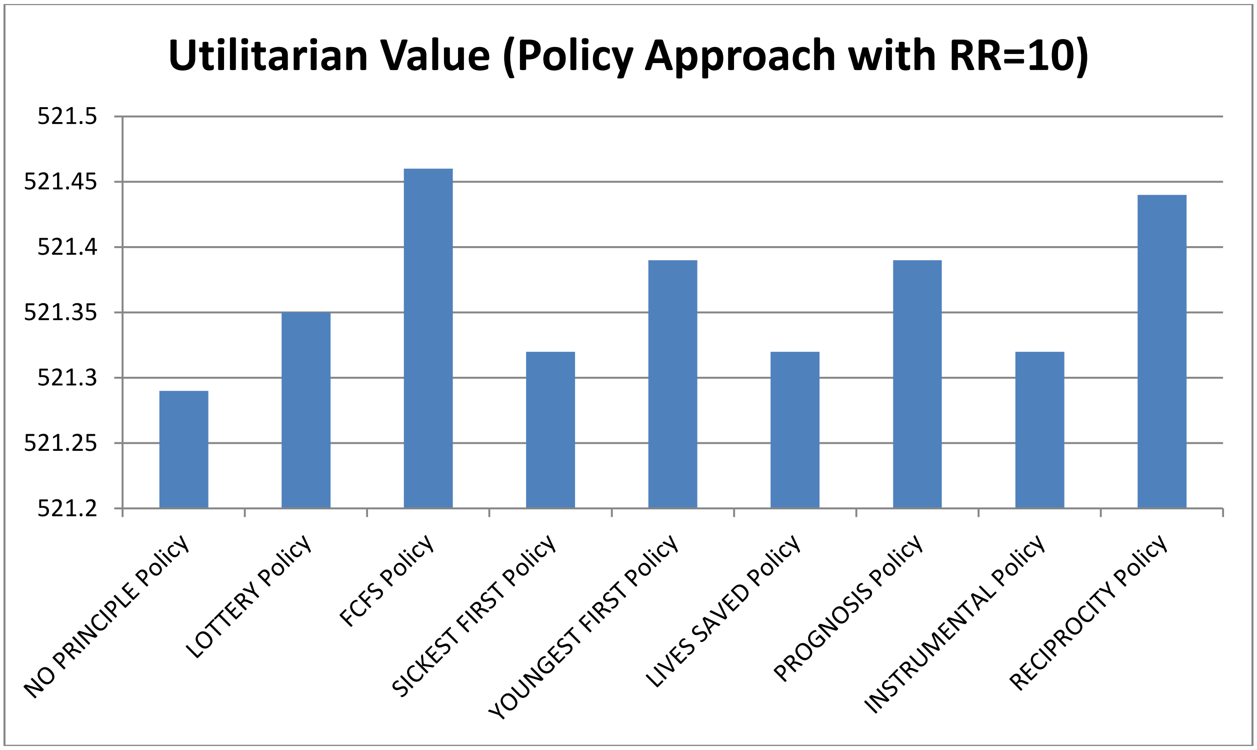

Figure VA-1c – Ambulance: Initial Utilitarian Allocation Value Policy Approach with RR=10

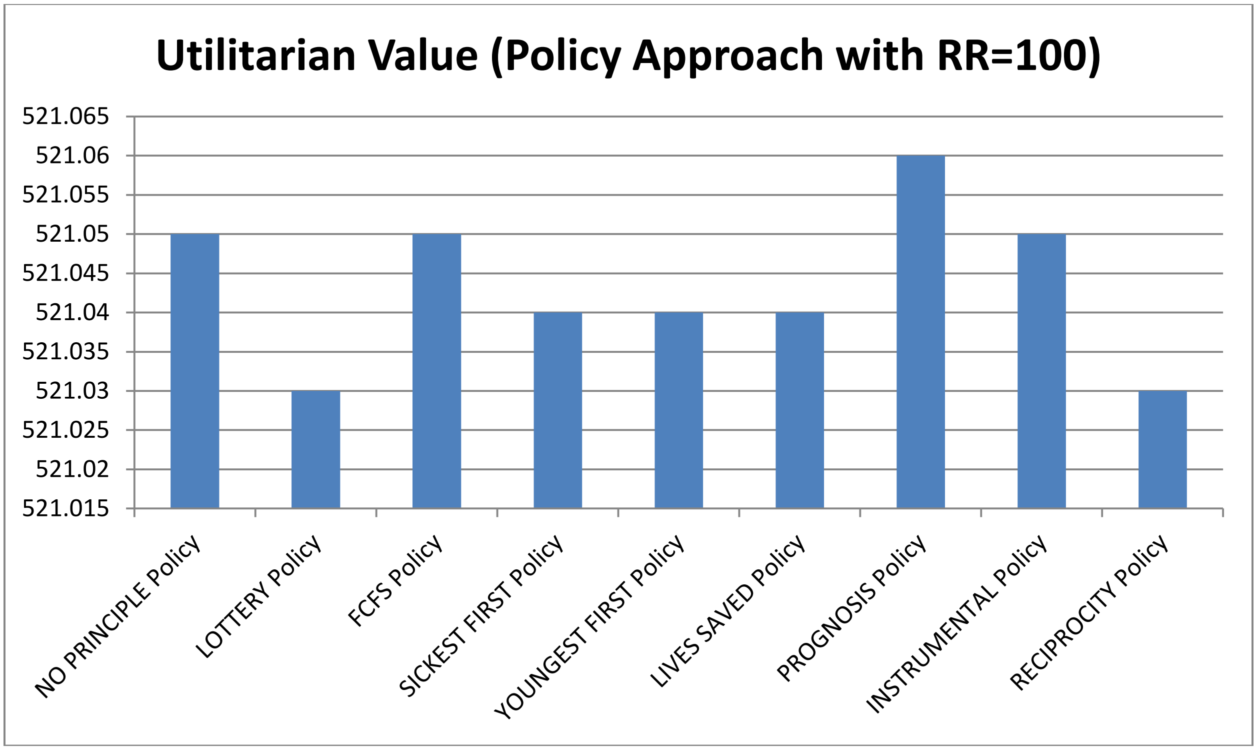

Figure VA-1d – Ambulance: Initial Utilitarian Allocation Value Policy Approach with RR=100

Table VA-1 – Ambulance: Average running time taken (in seconds) to complete an approach

Time Taken in Seconds	Principle	Policy (RR=1)	Policy (RR=10)	Policy (RR=100)
NO PRINCIPLE		0.0103660245	0.0915518492	0.9432083402
LOTTERY	0.0000803813	0.0100910939	0.0886819532	0.9202452037
FCFS	0.0000766149	0.0101563647	0.0880714489	0.9157133440
SICKEST FIRST	0.0000769943	0.0098438841	0.0891756518	0.9204586427
YOUNGEST FIRST	0.0000772686	0.0102546375	0.0888247954	0.9198152098
LIVES SAVED	0.0000650985	0.0098165237	0.0900167434	0.9310053305
PROGNOSIS	0.0000661255	0.0102069968	0.0929409075	0.9463240459
INSTRUMENTAL	0.0000768955	0.0105187661	0.0908367448	0.9412969757
RECIPROCITY	0.0000827859	0.0105653830	0.0915415193	0.9479914408

5.2 Output of Value to Decision Makers

We now move on to examine graphs which may assist decision makers in specifying social welfare functions and principles of allocation; the graphs also serve the purpose of demonstrating the relative value of the two primary approaches to MARA (principle based and policy based). As mentioned, we will return to deepen our discussion of the relative differences of these two approaches in Section 5.3.

5.2.1 Ambulance Stage Output

Figure VA-2 (a) and (d) provide additional valuable information for the decision maker. In these figures the y-axis represents the number of victims and the x-axis represents the approach taken. These figures are stacked bar graphs which categorizes by victims' severity and availability. This helps identify certain aspects of the allocation such as, for example, the number of victims with severity 2 that have been allocated a resource. To get a better idea of Figure VA-2 (a) we will look at the first bar, which is the lottery principle approach. For the 20 victims that have been allocated a resource there are 5.02 victims with severity 1, 4.82 victims with severity 2, 5.2 victims with severity 3, and 4.96 victims with severity 4. For the 10 victims without a resource (available) there are 2.58 victims with severity 1, 2.46 victims with severity 2, 2.51 victims with severity 3, and 2.45 victims with severity 4. The sickest first principle ensures that all the victims of severity 1 have a resource and continues with this strategy until no more resources available, which causes many victims with severity 4 to not have resources. The lives saved and prognosis principles seem to have an opposite strategy as there are a lot of victims of severity 4 with resources and a lot of victims of severity 1 without resources. The decision maker will need to consider both the cost of the allocation and how patients of different severities are handled in order to decide how to specify their preferences. Figure VA-2 (d) shows the difference between the policy based approaches. Under utilitarian social welfare these are fairly comparable. We will now move on to consider other social welfare functions.

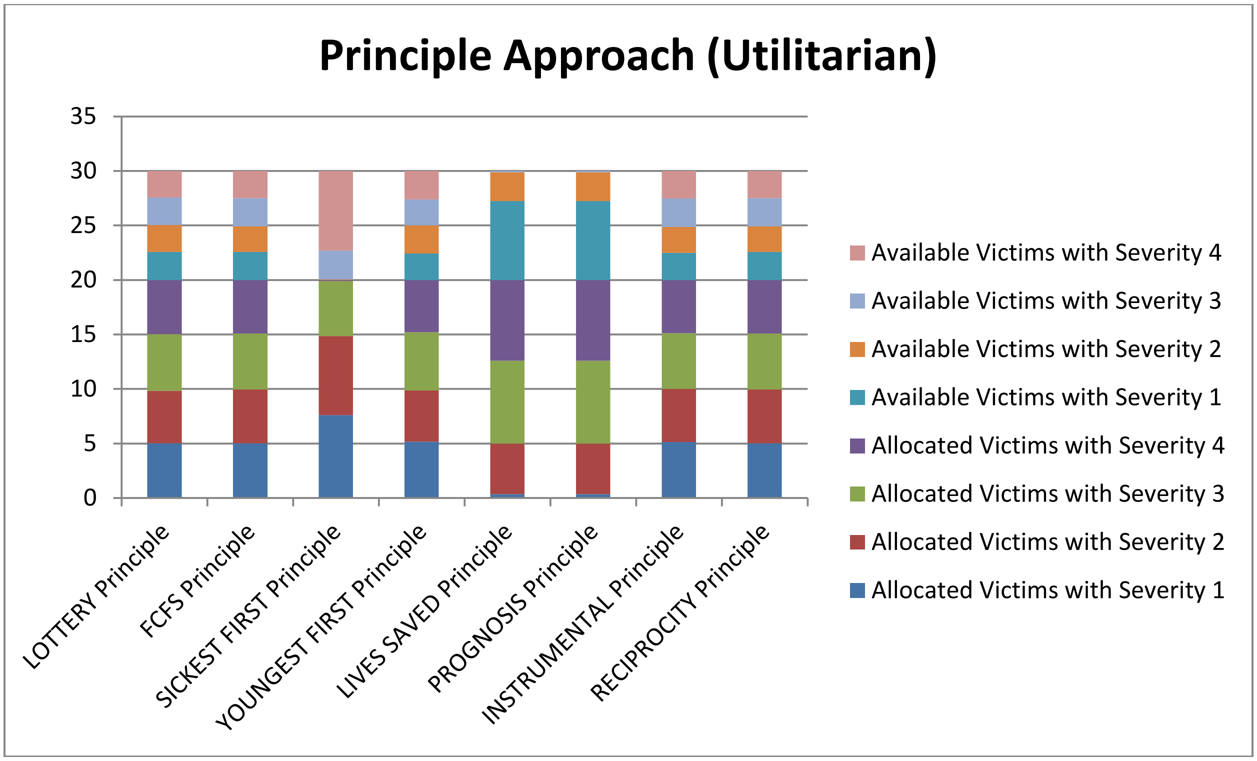

Figure VA-2a – Ambulance: Initial Allocation (Utilitarian Principle Approach)

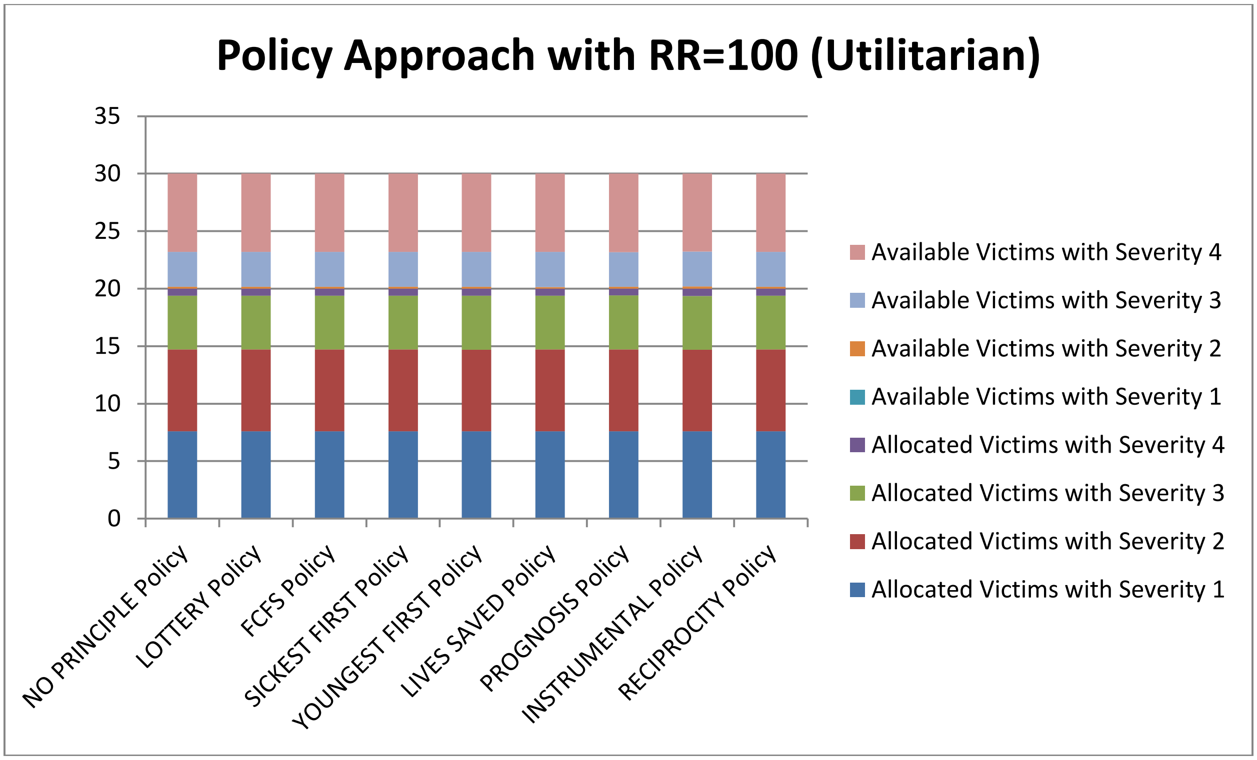

Figure VA-2d – Ambulance: Initial Allocation (Utilitarian Policy Approach with RR=100)

Figure VA-3 (a) and (d), Figure VA-4 (a) and (d) illustrate the Egalitarian social welfare function while Figure VA-5 (a) and (d), Figure VA-6 (a) and (d) illustrate the Nash Product social welfare function. The differences between the policy based approaches are somewhat more pronounced than in the Utilitarian case. Recall that the Egalitarian social welfare function focuses just the worst off individual and the Nash Product social welfare function is multiplicative.

In Figure VA-3 (d), the no principle, lives saved, and prognosis choices seem to have the best results in the policy approach. This may be due to this social welfare function focusing only on the worst off victim. The policy approach is not much better than the best of the principles of allocation approach (according to Figure VA-3 (a)). In fact, the sickest first principle approach has the best egalitarian value between both approaches. This is due to the algorithm of this principle, which focuses on giving the worst off victim the best possible resource. Egalitarian social welfare is more focused on a single individual in the society thus the policy based approach which reasons about the society as a whole do not offer significant advantage.

Looking at the allocations of resources to victims in Figure VA-4 (a) and (d) offers additional valuable input for the decision maker. For sickest first principle approach there are no victims of severity 1 without resources but there are few severity 4 victims with resources. In Figure VA-4 (d), we can that several of the policy based approaches, for example the no principle policy, perform only slightly worse than sickest first principle for attending to severity 1 victims but do far better in tending to the severity 4 victims as well.

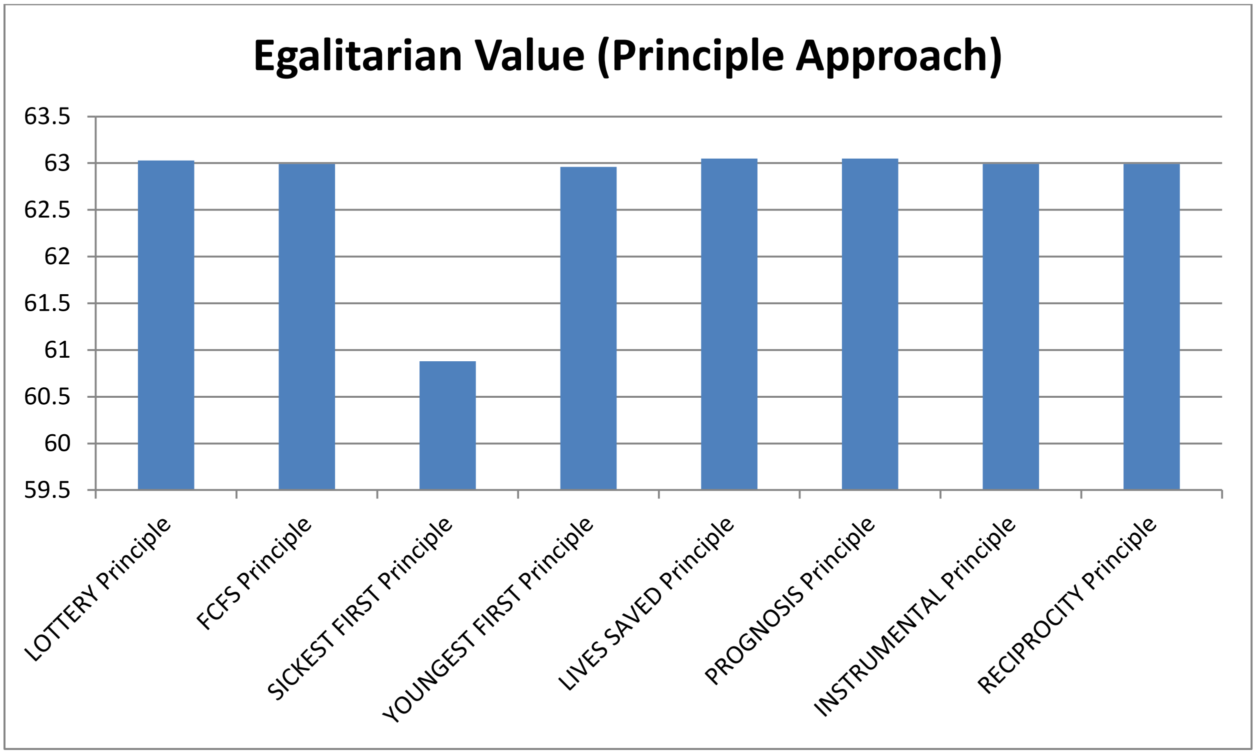

Figure VA-3a – Ambulance: Initial Egalitarian Allocation Value Principle Approach

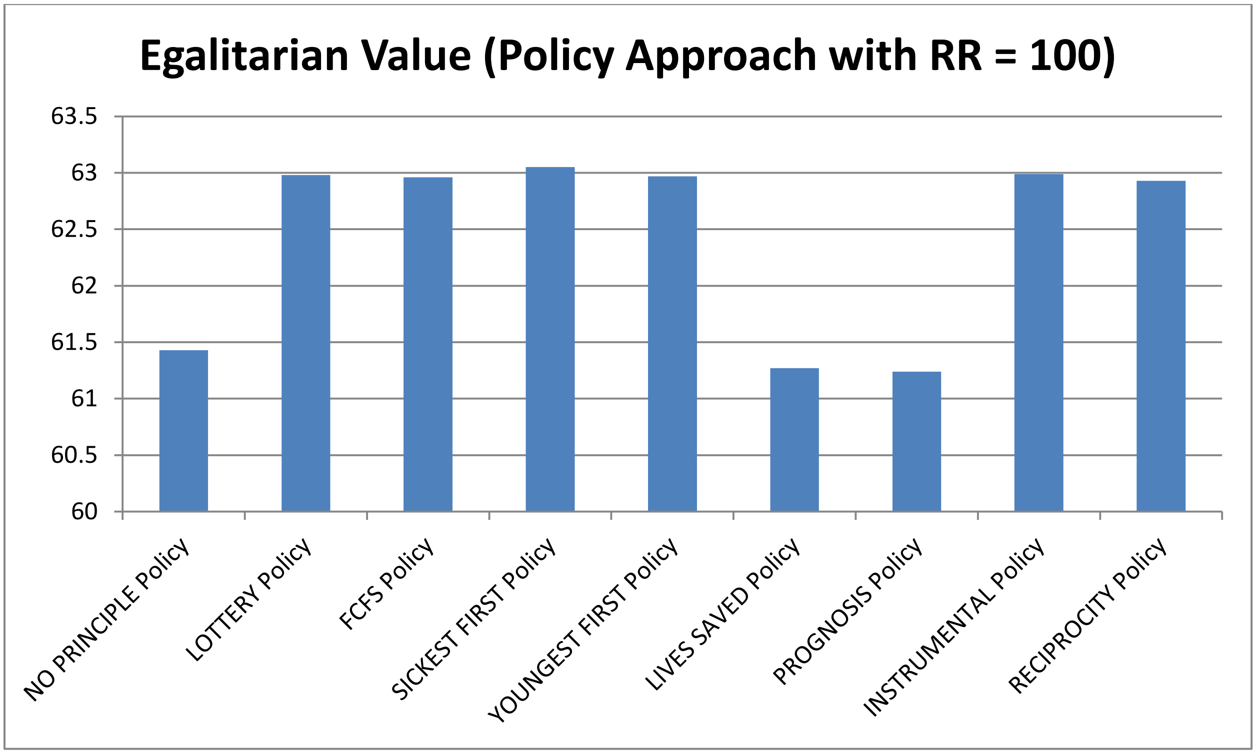

Figure VA-3d – Ambulance: Initial Egalitarian Allocation Value Policy Approach with RR=100

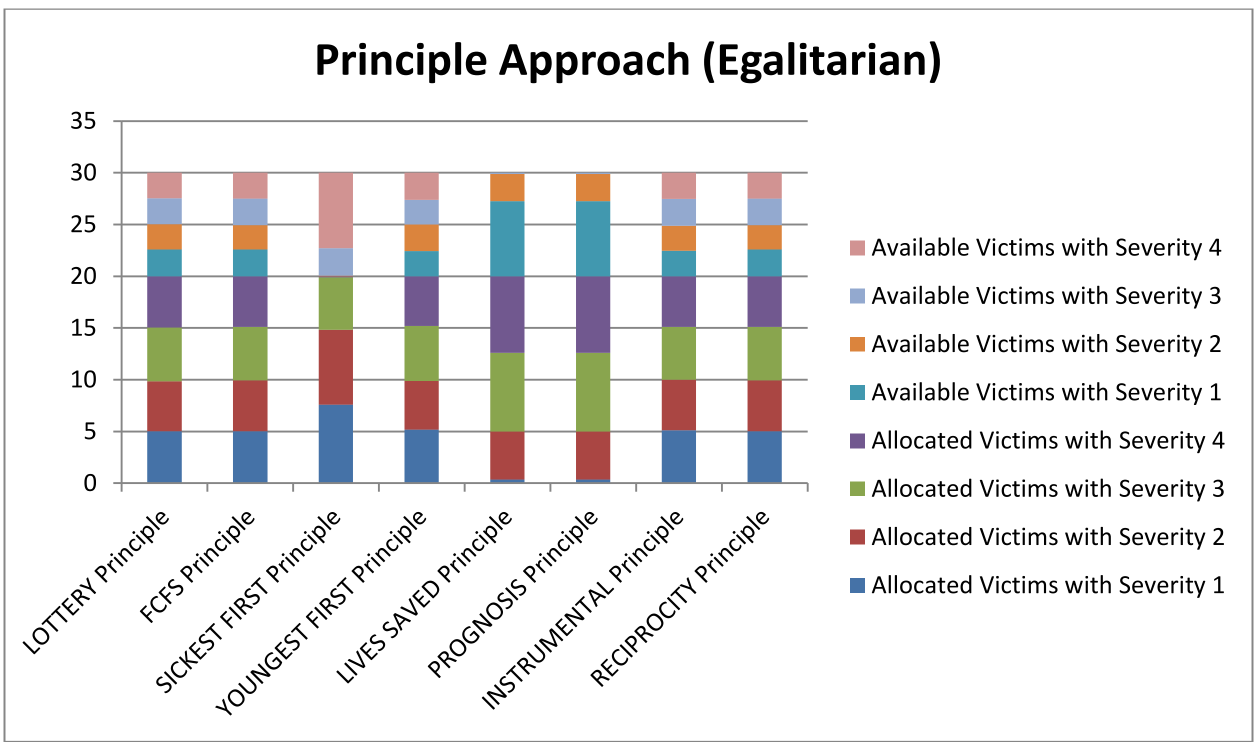

Figure VA-4a – Ambulance: Initial Allocation (Egalitarian Principle Approach)

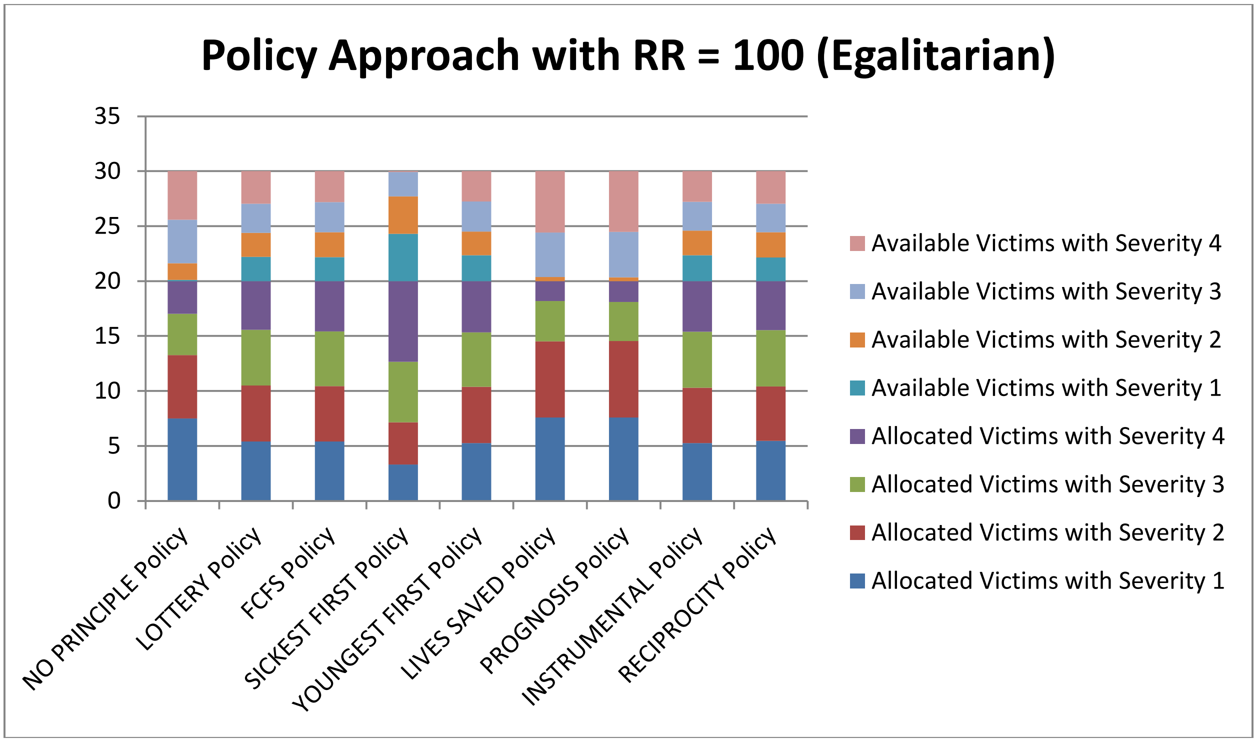

Figure VA-4d – Ambulance: Initial Allocation (Egalitarian Policy Approach with RR=100)

Looking at Nash Product social welfare function (in Figure VA-5 (a) and (d)) we note that there may be a huge difference in values since Nash Product is a multiplicative social welfare function. Note that each y-axis has a different range. Again the sickest first principle of allocation tends to do the best out of the principles of allocation approaches. However, some of the policy approaches (such as no principle and sickest first) tend to do better than all of the principle of allocation approaches. Moreover, the lives saved and prognosis results for both approaches were the worst.

Looking at Figure VA-6 (a) and (d) a decision maker will learn that certain options are not very desirable such as lives saved and prognosis principle (which do not address severity 1 patients very well). Comparing sickest first in Figure VA-6 (a) to no principle in Figure VA-6 (d) both do well with severity 1 patients but the no principle policy address severity 4 patients more effectively.

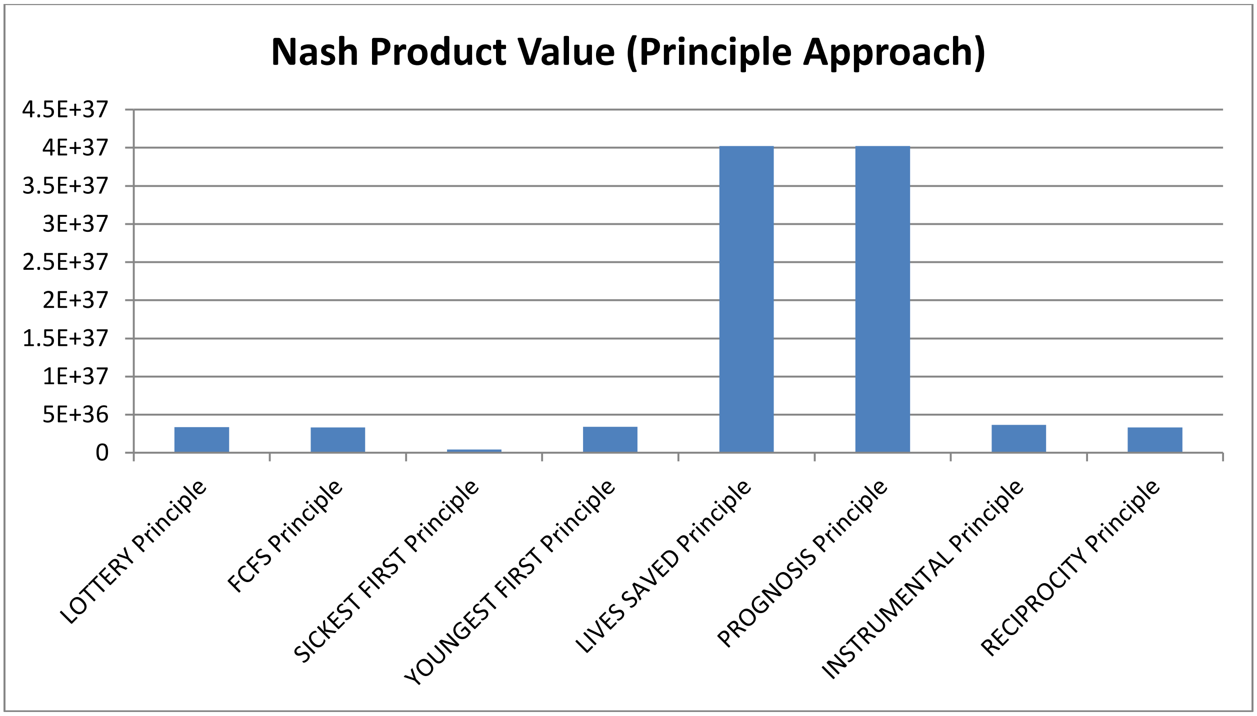

Figure VA-5a – Ambulance: Initial Nash Product Allocation Value Principle Approach

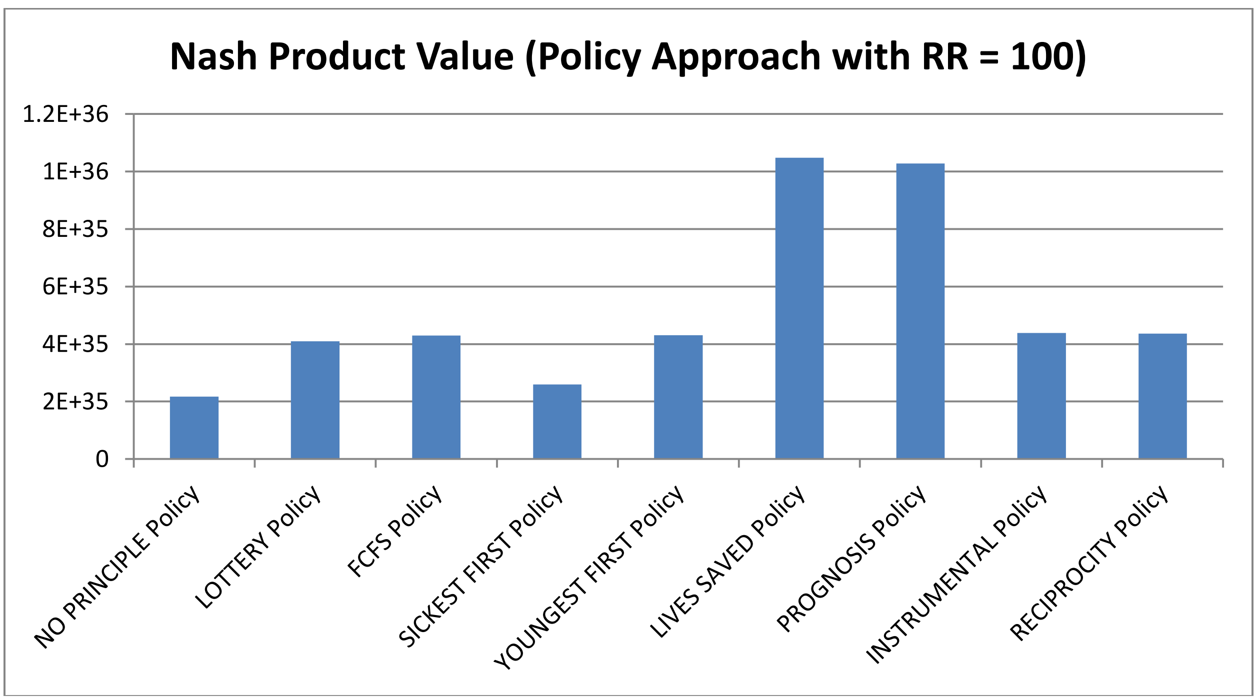

Figure VA-5d – Ambulance: Initial Nash Product Allocation Value Policy Approach with RR=100

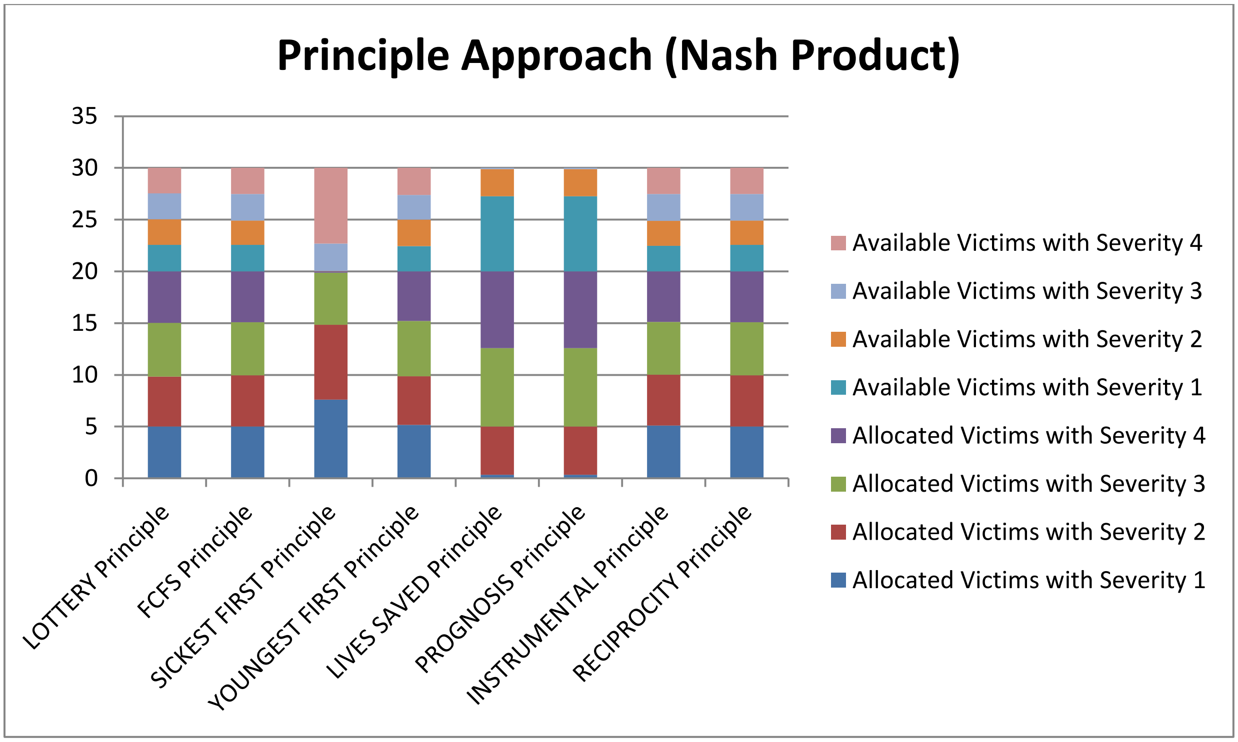

Figure VA-6a – Ambulance: Initial Allocation (Nash Product Principle Approach)

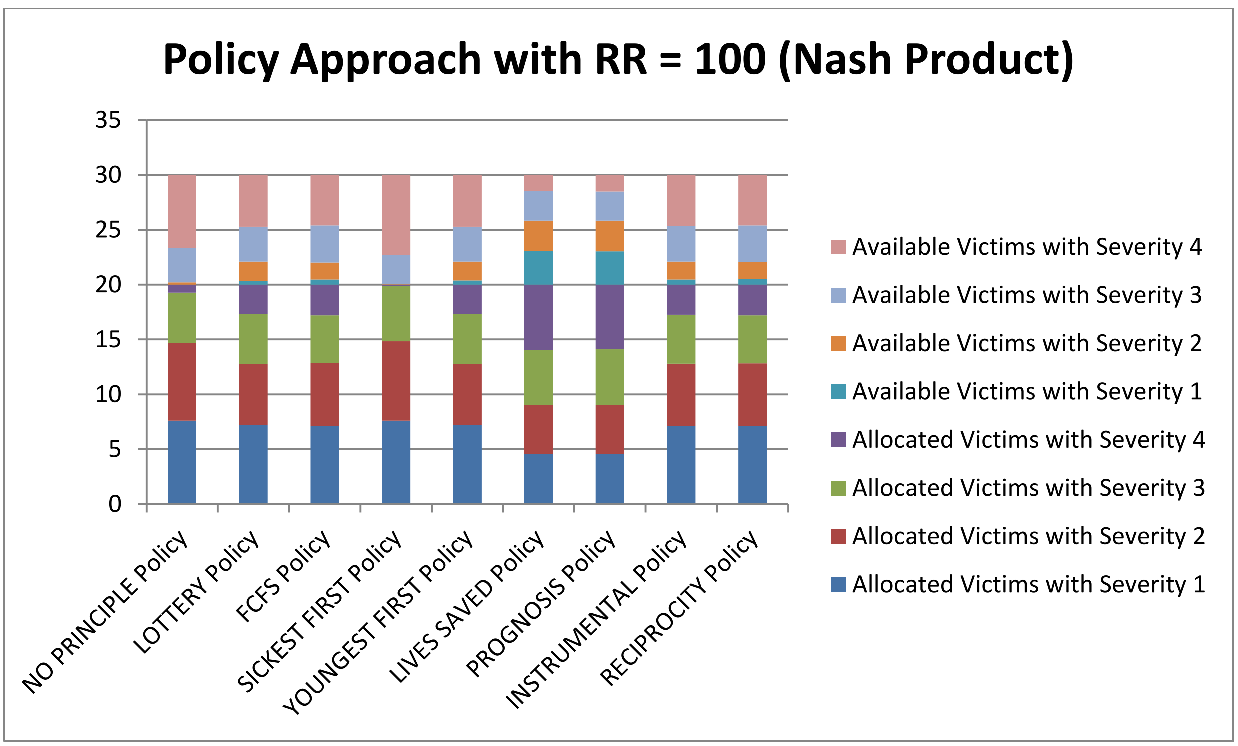

Figure VA-6d – Ambulance: Initial Allocation (Nash Product Policy Approach with RR=100)

So far the graphs that we have shown (Figures VA-1, VA-3, and VA-5) focus on displaying the total cost of an allocation (averaged over several runs) during the Initial phase of the Ambulance stage. Basically, which victims are going into which ambulances and to which hospitals has been decided. As time progresses two important events occur. The first is sensing, which may update parameter values and suggest a reallocation. For example, an ambulance headed for a particular hospital may be reassigned to a different hospital, if that allocation improves the cost. The other event that occurs is that the task of delivering the victim to the hospital may end, causing the ambulance to return to the site of the mass casualty incident to tend to new victims, who are then moved along to their assigned hospitals. The ideal is of course to make sure that all victims arrive at hospitals promptly, with the costs incurred for doing so kept low, overall (over the entire time period).

The first set of graphs examines how the global allocation value evolves, over time. After each time step, the value of the allocation is determined. Recall that this is a total cost calculation and thus how all the victims are doing is considered. The Cumulative Allocation value refers to the sum of the allocation value computed after each time step. Figure VA-7 (b) shows how the allocation value adjusts over time. The No-Policy curve serves to show the result of using our local search approach (where policies with principles have somewhat similar behaviors and are displayed in Appendix G); the remaining curves show the performance of MARA based on principles of allocation alone. We first see that No Policy and Sickest First have relatively lower cost values than the alternative choices. In Figure VA-8 (b) the Y-axis tracks how many of the runs out of the 100 that constitute the simulations have yet to be completed while the X-axis tracks the time. It is desirable to have fewer simulations still running: this means that all the victims have reached their hospitals. In Figure VA-8 (b) we see that No Principle fares the best; Sickest First is second best and the rest are not as strong. It is important to look at these results in the context of a tracking of the Cumulative Allocation value (Figure VA-10 (c)). Here we see that all of the policy approaches are superior: they have a lower total cost, after all the simulations have been completed. In particular, No Principle is better than Sickest First.

A decision maker could now view all the results regarding the allocation values and the number of simulations completed, over time, together with the cumulative allocation values in order to decide which options are preferable.

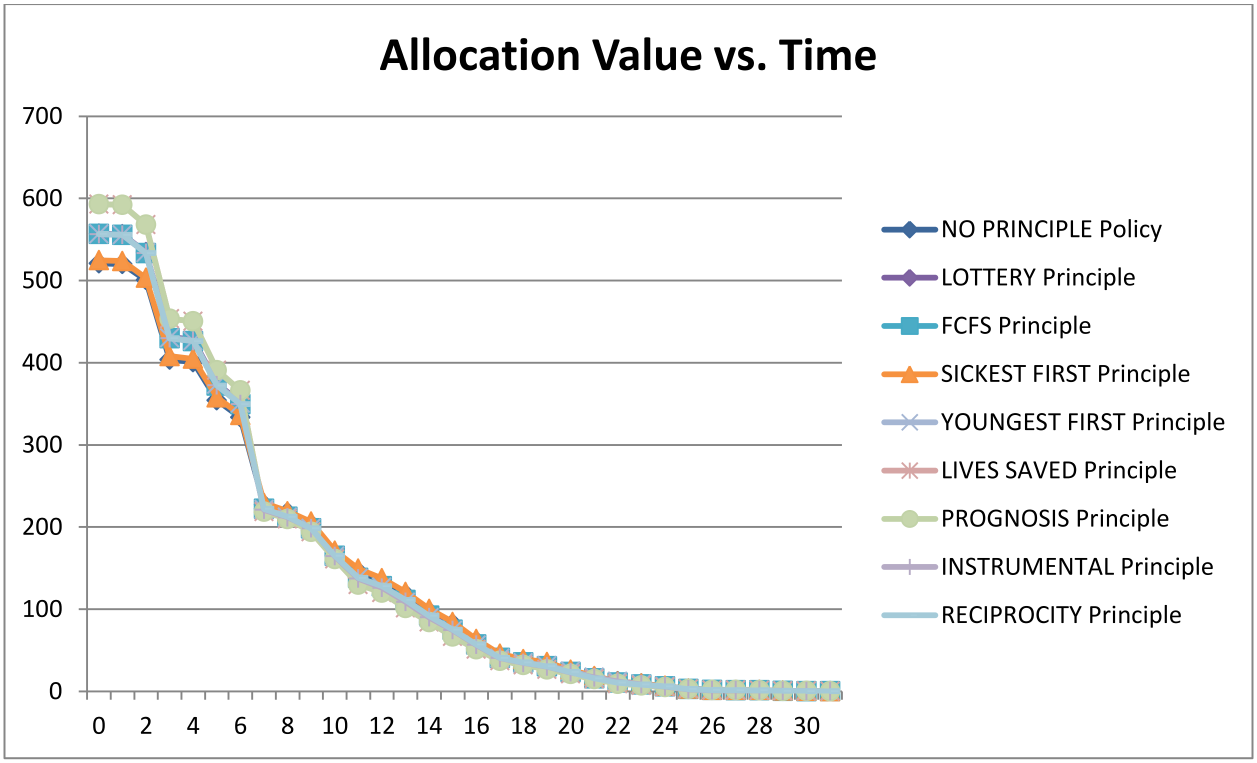

Figure VA-7b – Ambulance: Allocation Value over Time

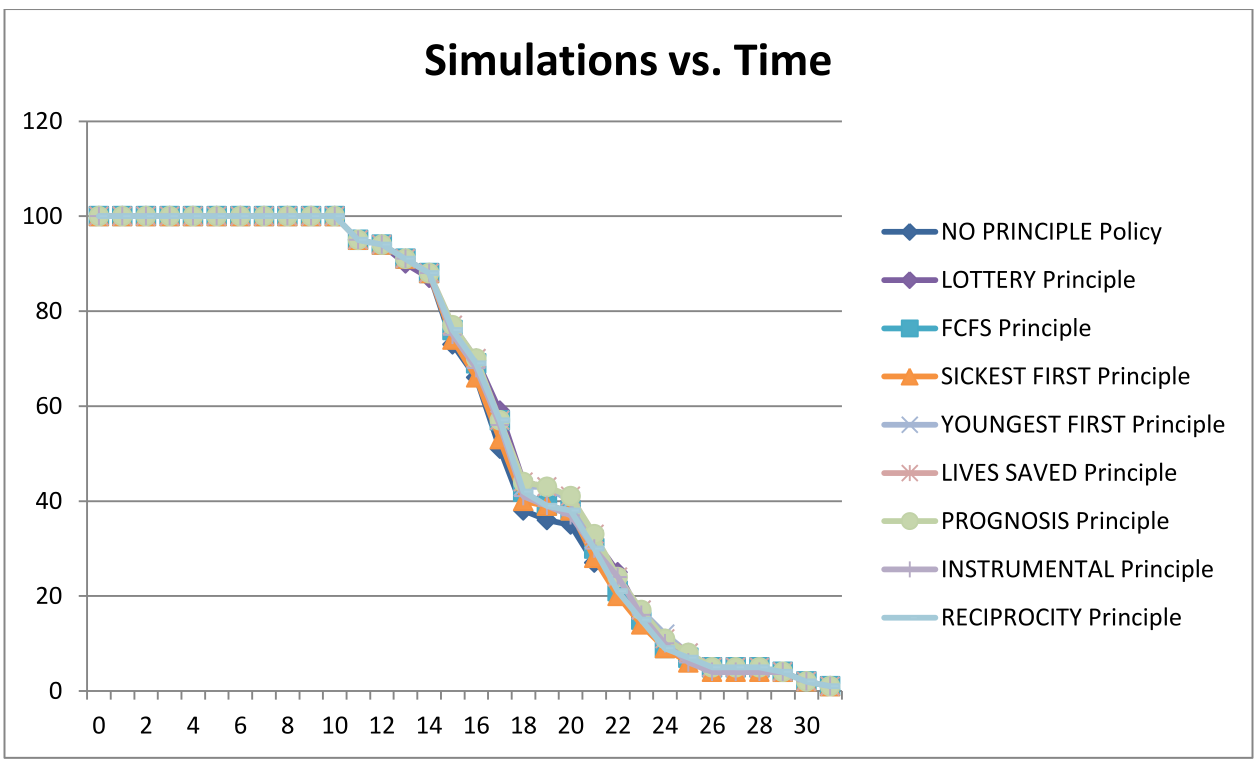

Figure VA-8b – Ambulance: Simulations over Time

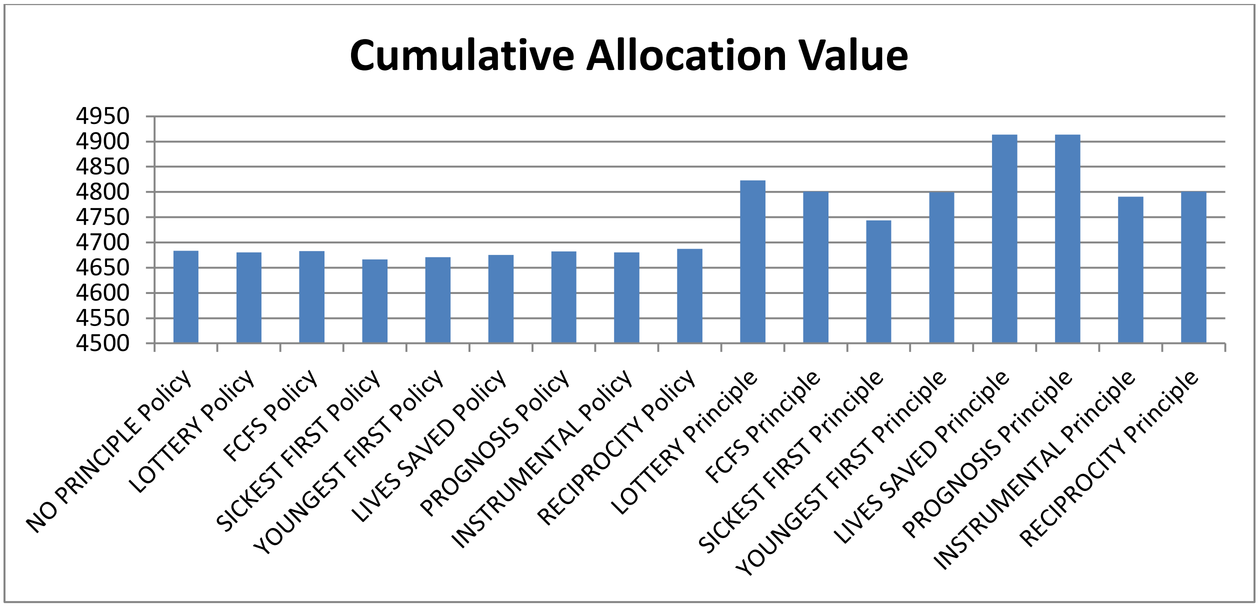

Figure VA-10c – Ambulance: Cumulative Allocation Value

The next set of graphs will assist the decision maker in setting the interval for sensing (e.g. after every two time steps). Sensing allows parameter values to be updated but in real scenarios, it may need to be done less frequently. The next figures show the relative difference between trying to sense every 2 time steps (Figure VA-11 (a)) compared to every 3 time steps (Figure VA-11 (b)) measured by tracking the allocation value, over time. We learn that when you sense less frequently, the overall amount of time it takes to achieve the overall cost of 0 is longer. We also notice more of a staircase approach for these graphs compared to Figure VA-7 (b) (which senses every time step). The difference between the curves in these two figures is also less pronounced than they were in Figure VA-7 (b) (i.e. the No Principle policy offers less of a gain over its competitors). Decision makers can decide how often to sense based on output such as this.

The last series of graphs for the Ambulance stage examines possible choices for the length of Buffer Time. Figure VA-15 (a) has buffer of 2 time steps and senses every time step while Figure VA-15 (b) has the same buffer length but senses every two time steps. We first compare Figure VA-15 (a) with Figure VA-7 (b) (which senses every time step but does not have a Buffer Time). Because one has to wait with a Buffer Time, the amount of time until you've completed the simulation increases which means the overall allocation time also increases. Comparing Figure VA-15 (a) with Figure VA-15 (b) we see the effects see the effects of sensing every other time step: a staircase approach which was evident in the previous set of graphs.

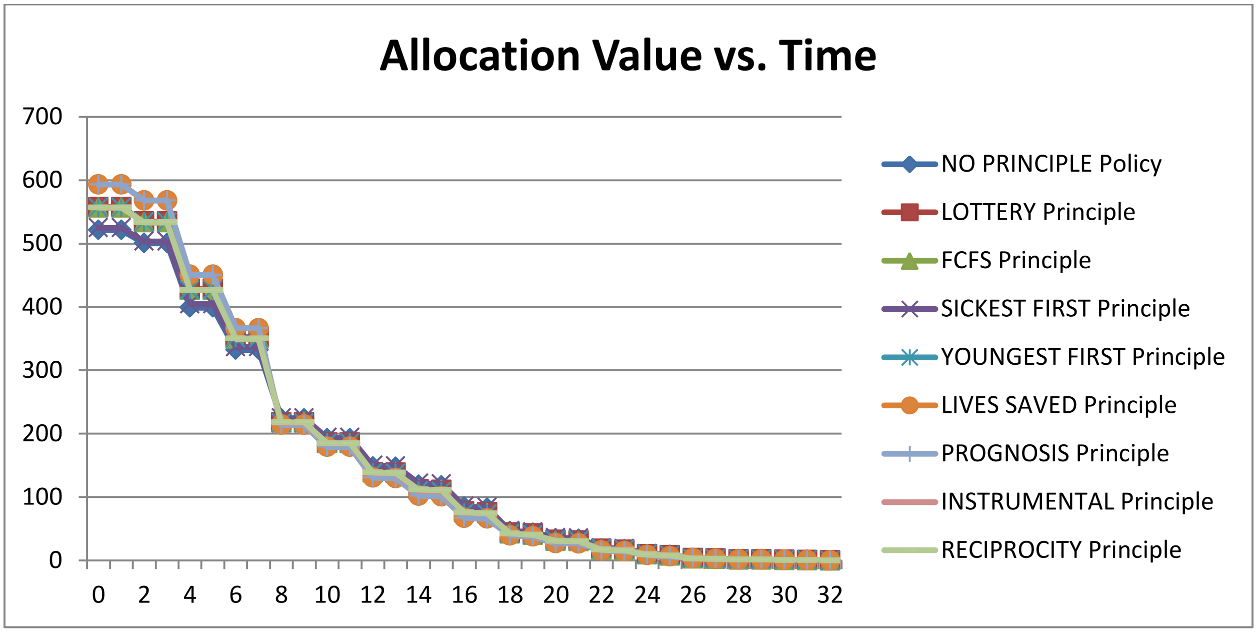

Figure VA-11a – Ambulance: Allocation Value over Time (Sensing=2)

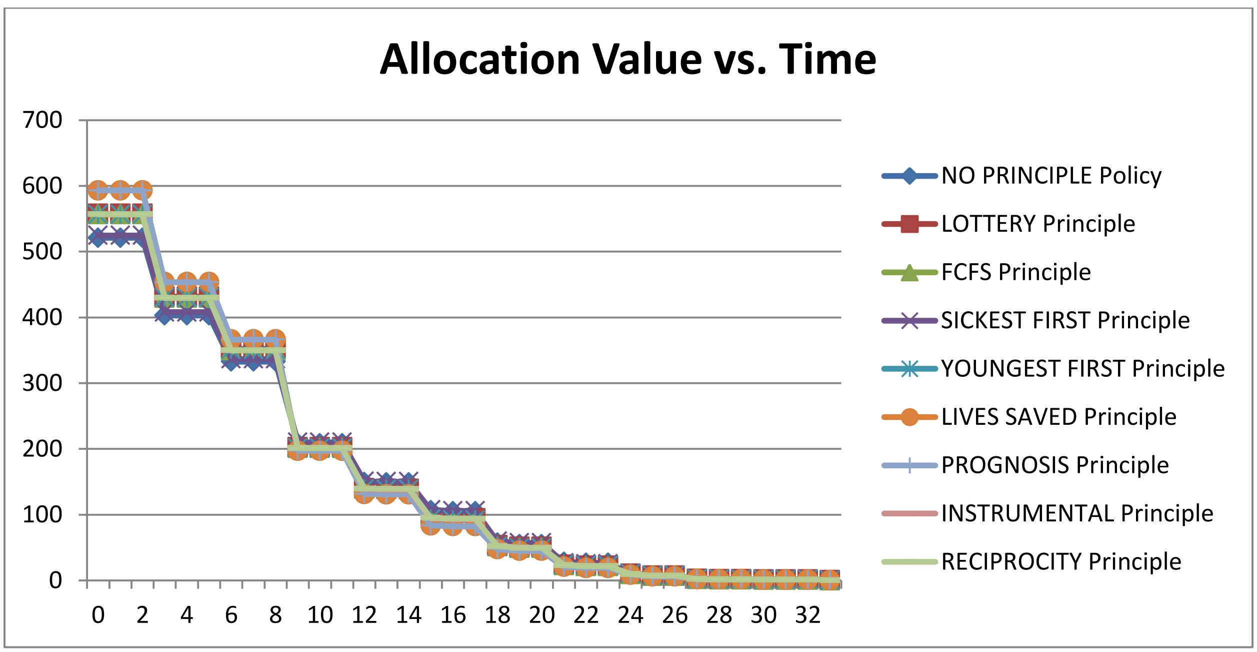

Figure VA-11b – Ambulance: Allocation Value over Time (Sensing=3)

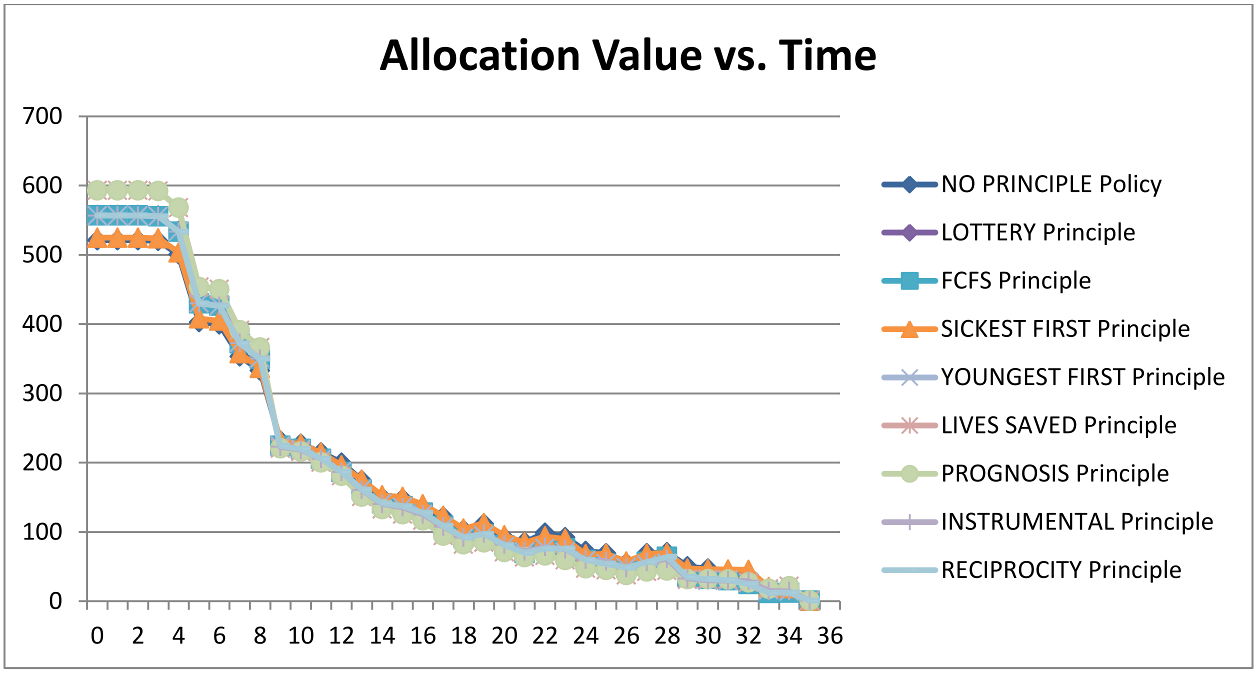

Figure VA-15a – Ambulance: Allocation Value over Time (Sensing=1, Buffer=2)

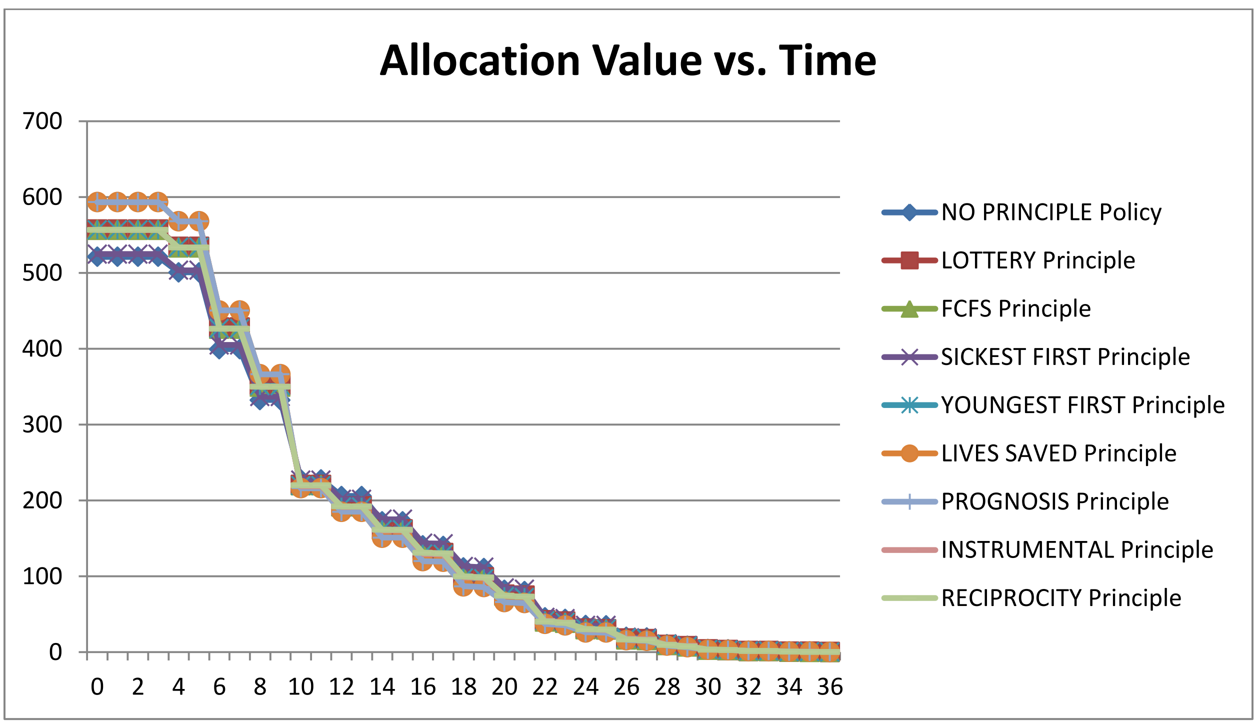

Figure VA-15b – Ambulance: Allocation Value over Time (Sensing=2, Buffer=2)

5.2.2 Hospital Stage Output

In this section we will show several graphs which are counterparts to the ones we showed in section 5.2.1 but this time for the Hospital stage. Note that it is important to do so because the Hospital stage uses a different total cost function and therefore the outcomes (both the allocation values and the comparisons) may be different. There are also some graphs in this section that are unique to the Hospital stage. In particular, some graphs will be tracking allocation value per patient (over time or cumulative).

The motivation to using these graphs is due to the difference between these stages. In the Ambulance stage, when a victim is done using their resource, then they no longer need any more resources. In contrast, in the Hospital stage, when a patient is done using their resource they may not be discharged from the hospital and now require another resource. How we model the stopping point is as follows: when a good severity (severity 4) patient is done with its final resource then that patient will have a discharge severity (severity 5) and will be discharged from the hospital since the patient no longer requires a resource.

As specified in the Ambulance stage, the first set of graphs that we will take a look at is the initial allocation Utilitarian values. Figure VH-1 (a) is the principle of allocation approach and Figure VH-1 (d) is the policy approach with 100 random restarts. Note that Table VH-1 shows that the average running time (computed over the 100 runs) is still less than 1 second, which makes using 100 random restarts a viable option. In the similar pattern seen in the Ambulance stage, the sickest first principle has the best allocation value for the principle approaches and that all of the policy approaches have a better allocation value than the principle approaches. Looking at Figures VH-2 (a) and (d) (stacked bar graphs which categorizes by patients' severity and availability for the principle and policy approaches) we note that the sickest first principle approach has been allocated the same as each of the policy approaches. This emphasizes that even though the same people for each severity level may be allocated a resource for the sickest first principle that the policy approaches allocates better globally such that there is a lower cost.

Since we decided in the Ambulance stage that we will focus on the Utilitarian social welfare function, the initial results for the Egalitarian and Nash Product have been moved to Appendix G. For the Egalitarian social welfare, sickest first principle and the no principle policy had the best initial Egalitarian value. For the Nash Product social welfare, all of the policy approaches were better than the sickest first principle (the best principle approach).

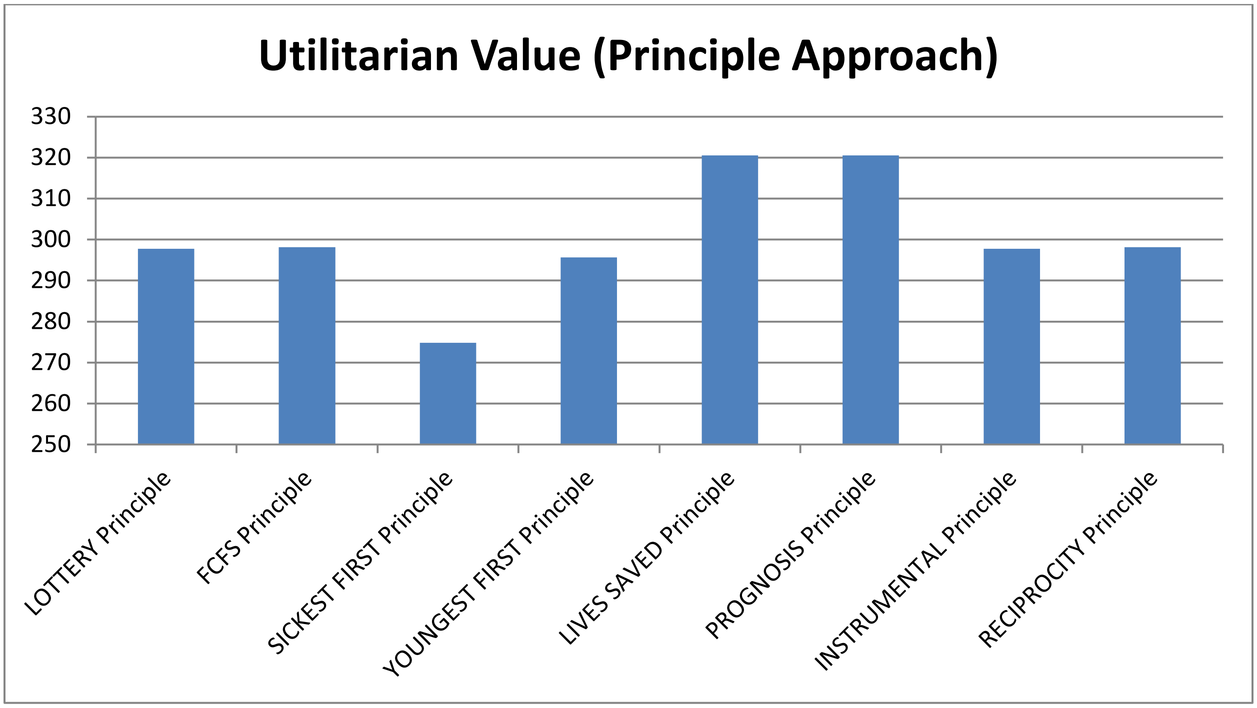

Figure VH-1a – Hospital: Initial Utilitarian Allocation Value Principle Approach

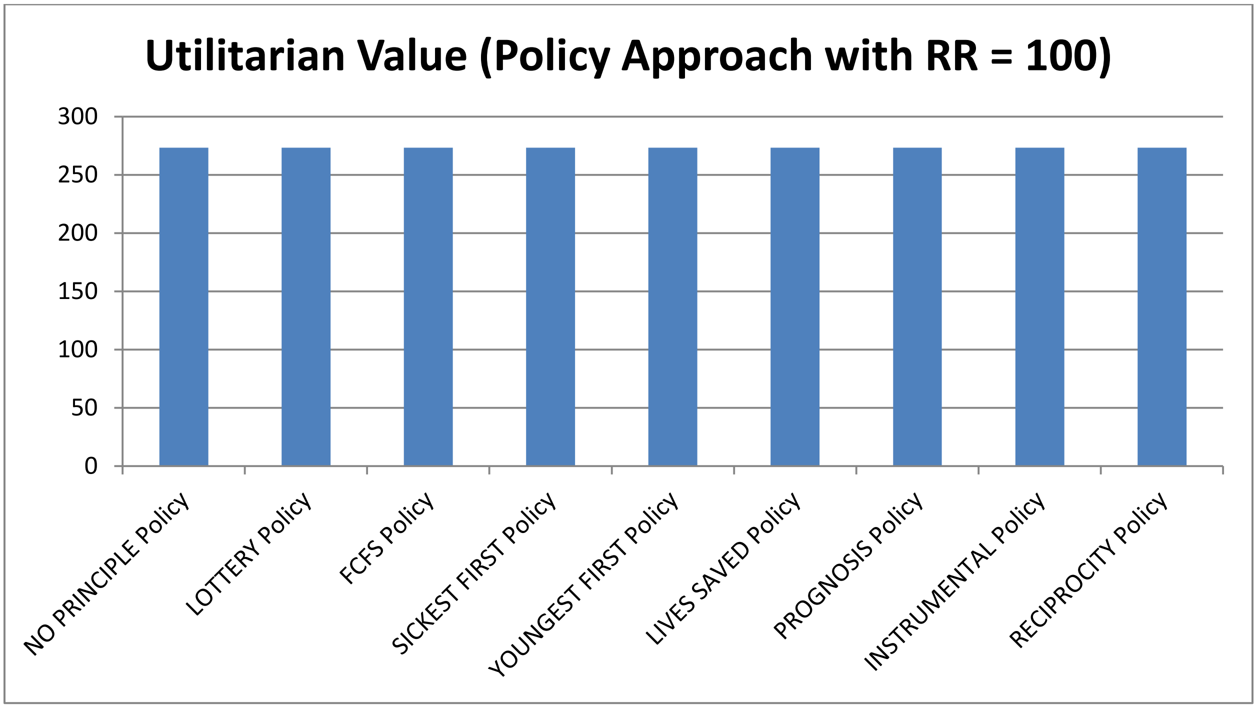

Figure VH-1d – Hospital: Initial Utilitarian Allocation Value Policy Approach with RR=100

Table VH-1 – Hospital: Average running time taken (in seconds) to complete an approach

Time Taken in Seconds	Principle	Policy (RR=1)	Policy (RR=10)	Policy (RR=100)
NO PRINCIPLE		0.0000870500	0.0007913941	0.0088430344
LOTTERY	0.0000254660	0.0000939770	0.0008013762	0.0082970964
FCFS	0.0000270702	0.0000918721	0.0008216661	0.0083063418
SICKEST FIRST	0.0000286201	0.0001014557	0.0008000462	0.0082534549
YOUNGEST FIRST	0.0000281481	0.0000955142	0.0007882111	0.0083499577
LIVES SAVED	0.0000266811	0.0000891293	0.0008557270	0.0084598489
PROGNOSIS	0.0000261644	0.0000895918	0.0008609128	0.0084844697
INSTRUMENTAL	0.0000277368	0.0000897608	0.0008873067	0.0085190218
RECIPROCITY	0.0000276602	0.0000881949	0.0008600963	0.0085009070

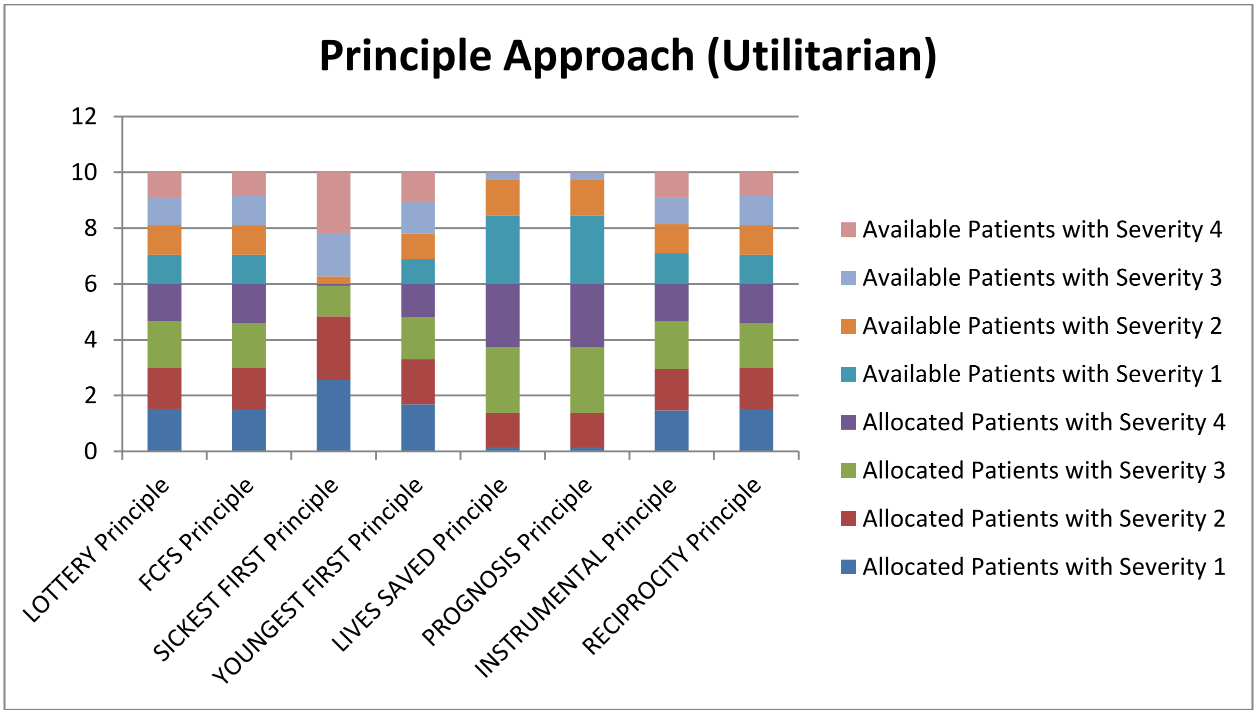

Figure VH-2a – Hospital: Initial Allocation (Utilitarian Principle Approach)

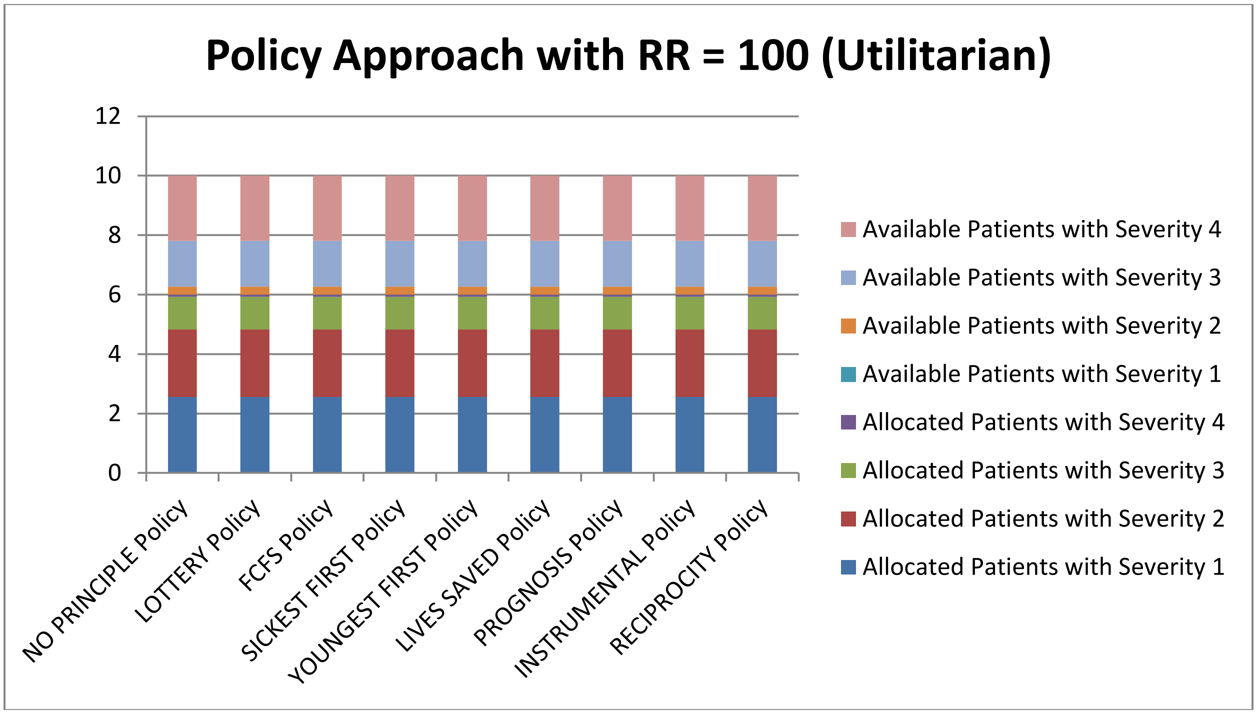

Figure VH-2d – Hospital: Initial Allocation (Utilitarian Policy Approach with RR=100)

Going beyond the Initial phase, we now show various graphs for the Update phase. In this hospital stage, we simulate 10 patients who need resources at a hospital. Each arrives with a randomly assigned severity level and a randomly assigned need for a procedure. The hospital is simulated as being able to provide 6 resources (where each resource is in fact a bundle that would be assigned to patient). The type of each resource (A, B, C, D, E or F) is set randomly.

For each procedure, we simulate how long it will take for a patient to be done with the resource that it is assigned, to finish the procedure. This is done based on a pre-set value for how long each procedure should take with that resource being assigned to handle it (e.g. a resource with that particular skill level).

As each new time tick proceeds, at some point the patient will be simulated as completing its procedure. At this point, their severity level is modeled as improving (increased by 1 from its previous level). As explained earlier, a level of 5 represents a patient who can now be discharged and no longer needs resources. Otherwise, the patient is modeled as requiring additional resources and becomes one of the patients to whom resources must be allocated, at the next time tick.

In order to view the allocation value over time, similarly to what we did in the Ambulance stage, we simply compare a no principle policy with the various principles (see Figure VH-7 (b)). Again the best performance is one which achieves lower values and reaches these values earlier over time (Figure VH-8 (b)). When comparing the allocation values of a no principle policy approach with the principle of allocation approaches the differences are noticeable. No principle policy and sickest first principle tend to be the best approaches during the first few time steps. After the initial time steps, the sickest first principle and no principle policy have the worst allocation values during the middle time steps. Eventually the allocation values will be best again for sickest first principle and no principle policy until no more simulations are performed and the allocation values are 0 since every patient has been discharged from the hospital. From these results, it seems that the sickest first principle is most similar to the policy approaches. The no principle policy approach tends to be better than the sickest first principle for most of the time. While this is positive, we should consider the other graphs before claiming that the policy approaches are the best.

Looking at Figure VH-8 (b), when comparing the no principle policy approach with the principle of allocation approaches the variations are more noticeable. The numbers tend to be fairly similar during the early time steps but no principle policy and sickest first principle have the fewest simulations running during the middle time steps. This leads to these approaches completing all their simulations before the others. All the policy approaches and the sickest first principle approach complete all their simulations at the 30th time step. This is followed by the lottery, youngest first, lives saved and prognosis principles completing all their simulations at the 36th time step. The remaining principles complete all their simulations at the 38th time step.

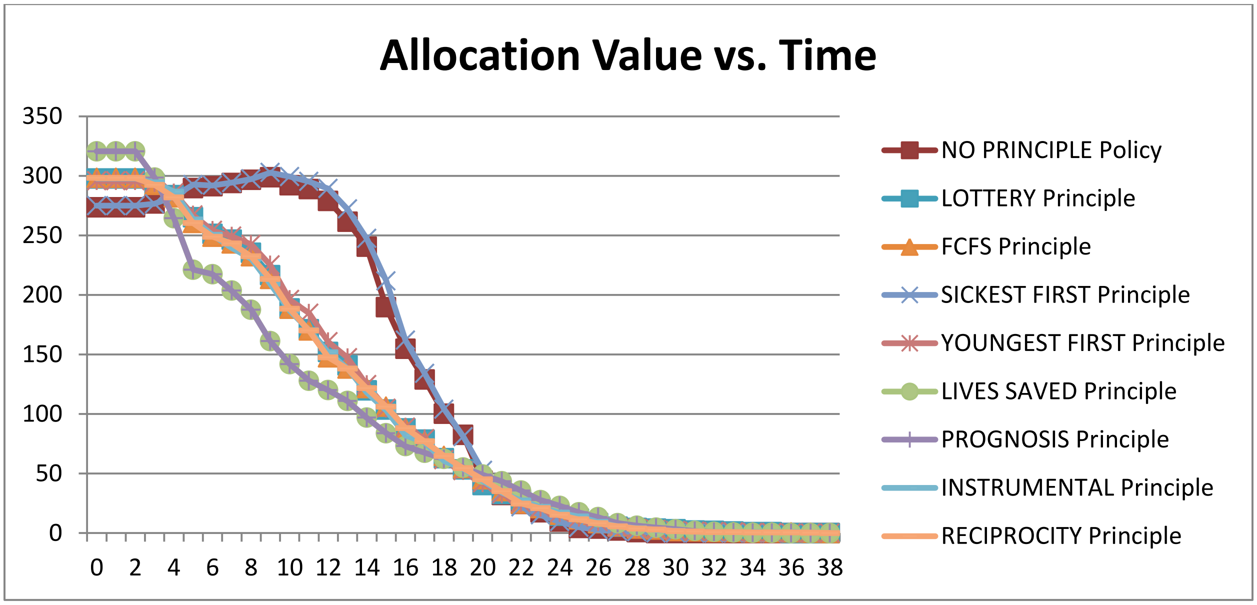

Figure VH-7b – Hospital: Allocation Value over Time

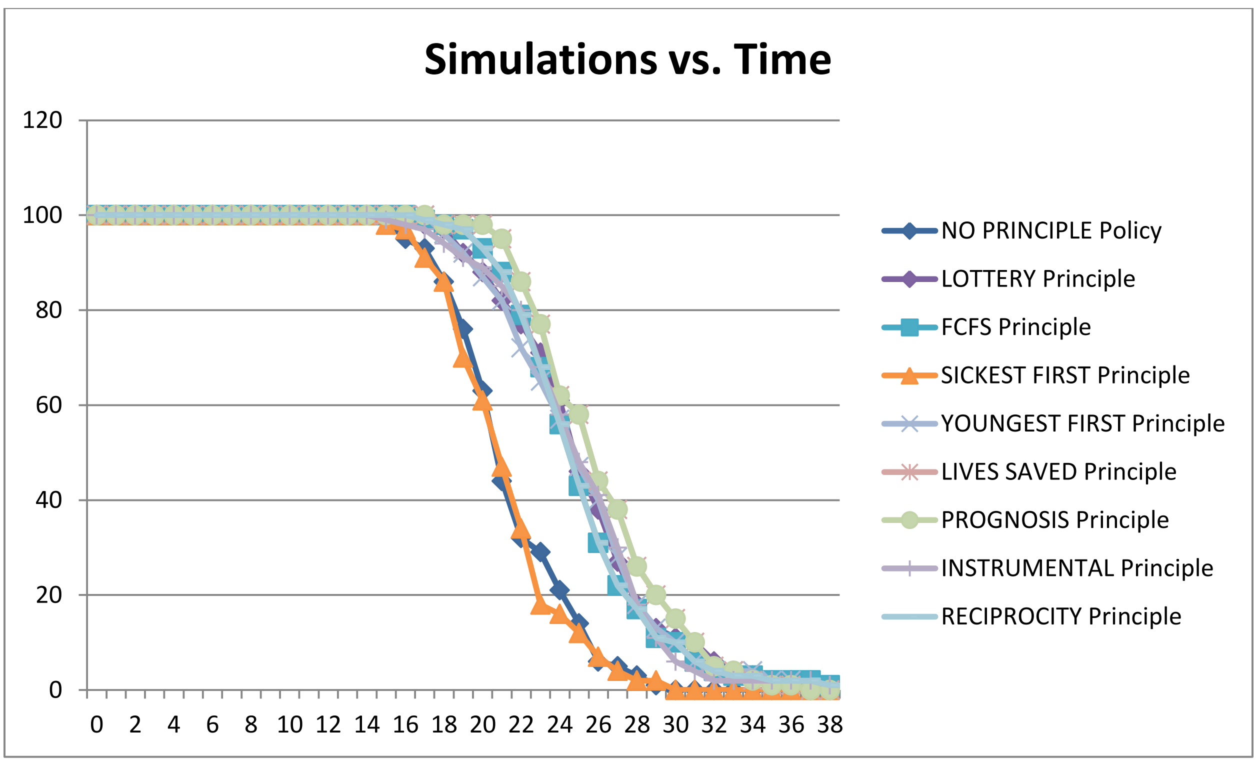

Figure VH-8b – Hospital: Simulations over Time

In the graphs that follow, we examine what will happen to patients over time. Decision makers should prefer choices where there are fewer patients in the hospital and also where all patients are finished with their resources earlier. What we see in Figure VH-9 (b) is the following. For prognosis and lives saved principles, the severity 4 patients are given priority and therefore will leave the hospital after receiving their resources (in a relatively short period of time), however, this leaves the severity 1 patients waiting longer to be handled and discharged from the hospital. On the other hand, the no principle policy and the sickest principle will begin with more patients remaining in the hospital (they are handling more severe patients who require more time to end their consumption of their resources); however these strategies benefit because the time point at which they have completed processing all of their patients is earlier.

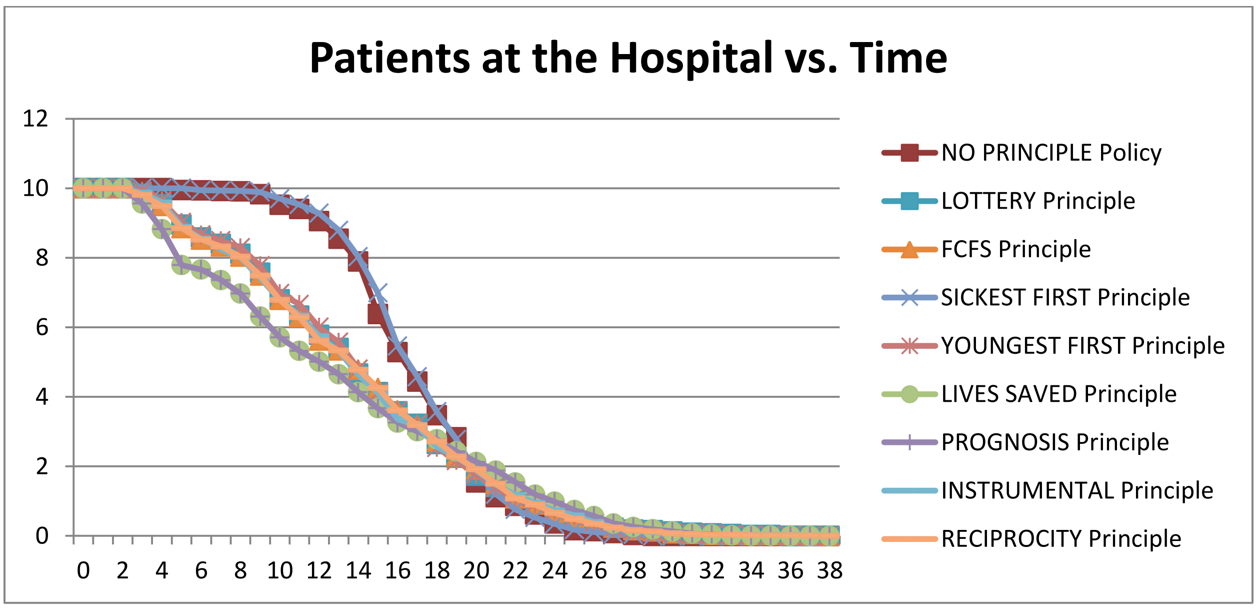

Figure VH-9b – Hospital: Number of Patient at the Hospital over Time

In Figure VH-10 (b), we are examining the allocation value per patient over time. The most dramatic pattern that we observe here is that the no principle policy and sickest first principle reach a value of 0 at time point 29 while the other approaches maintain a higher overall cost. The drop in the cost at this time point (and initially lower costs) will make up for having a higher cost during the middle time points. It is also important for the decision makers to evaluate the cumulative allocation value per patient, which adds the allocation value per patient for each time step (in Figure VH-10 (c)). Even though sickest first is the best of the principles, all of the policy approaches are considerably better.

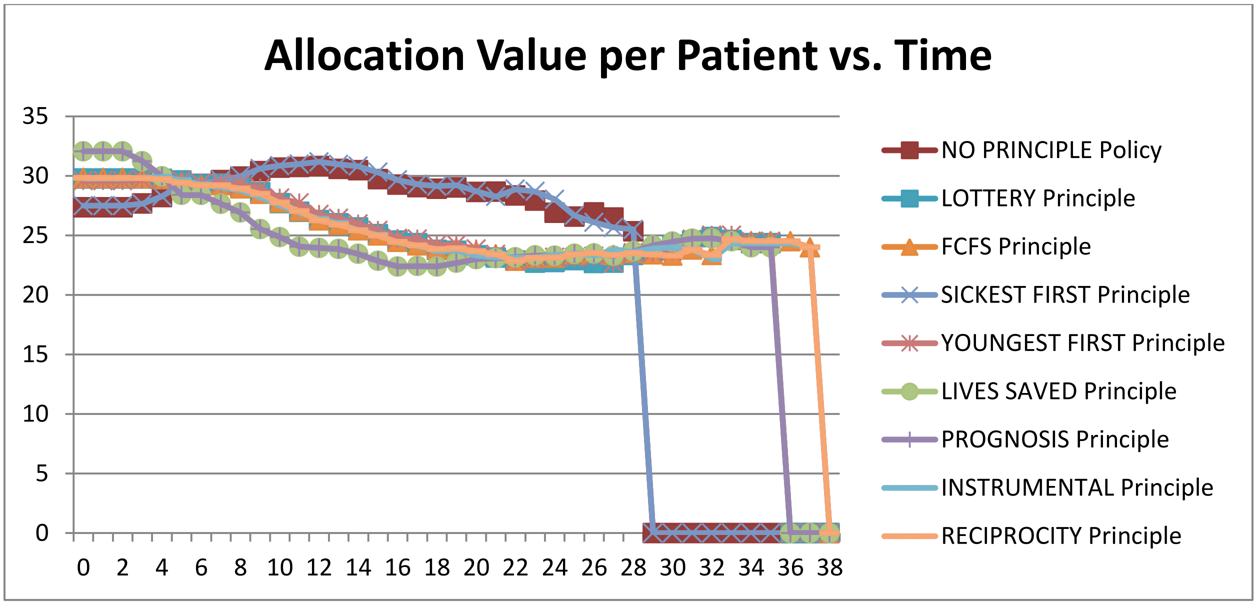

Figure VH-10b – Hospital: Allocation Value per Patient over Time

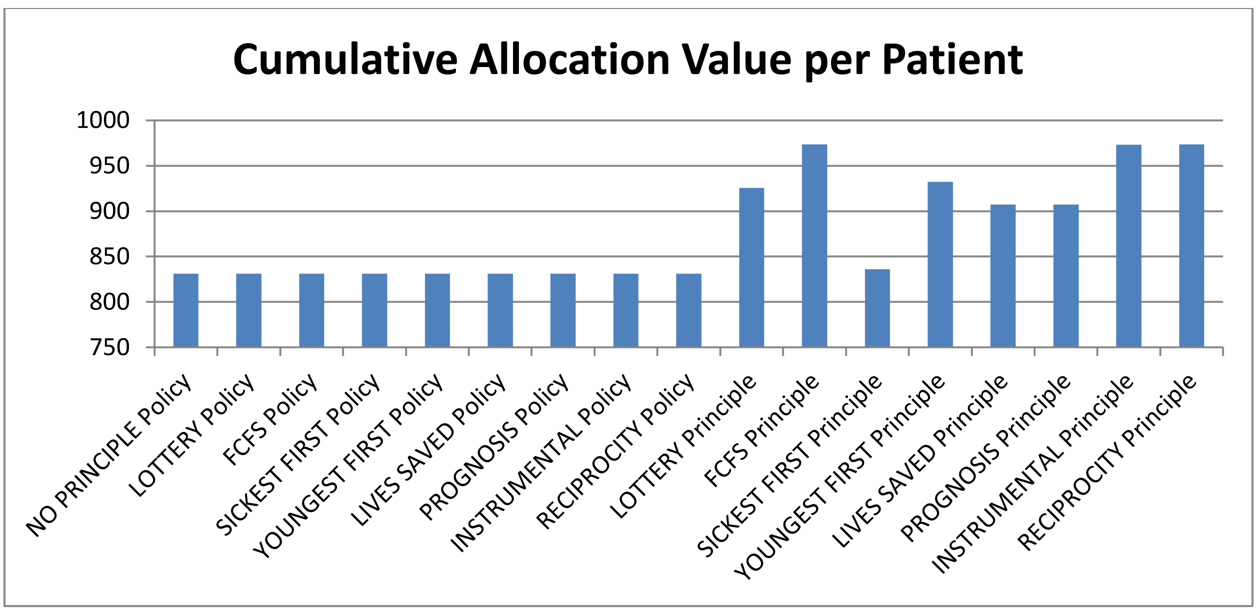

Figure VH-10c – Hospital: Cumulative Allocation Value per Patient

We conclude with a look at choices for sensing frequency and time buffer values. As stated in the previous stage, these graphs will assist decision makers in setting these values. The next figures take a look at the allocation value per patient over time, where Figure VH-14 (a) senses every 2 time steps while Figure VH-14 (b) senses every 3 time steps. Again, we see that sensing less frequently causes a longer amount of time, which will increase the costs and create more of a staircase look to these graphs.

The last series of graphs for the Hospital stage examines possible choices for the length of Buffer Time. Examining the allocation value per patient over time, Figure VH-18 (a) has buffer of 1 time steps and senses every time step while Figure VH-18 (b) has the same buffer length but senses every two time steps. We first compare Figure VH-18 (a) with Figure VH-10 (b) (which senses every time step but does not have a Buffer Time). Because one has to wait with a Buffer Time the amount of time until you've completed the simulation increases which means the overall allocation time also increases. Comparing Figure VH-18 (a) with Figure VH-18 (b) we see the effects of sensing every other time step: a staircase approach which was evident in the previous set of graphs.

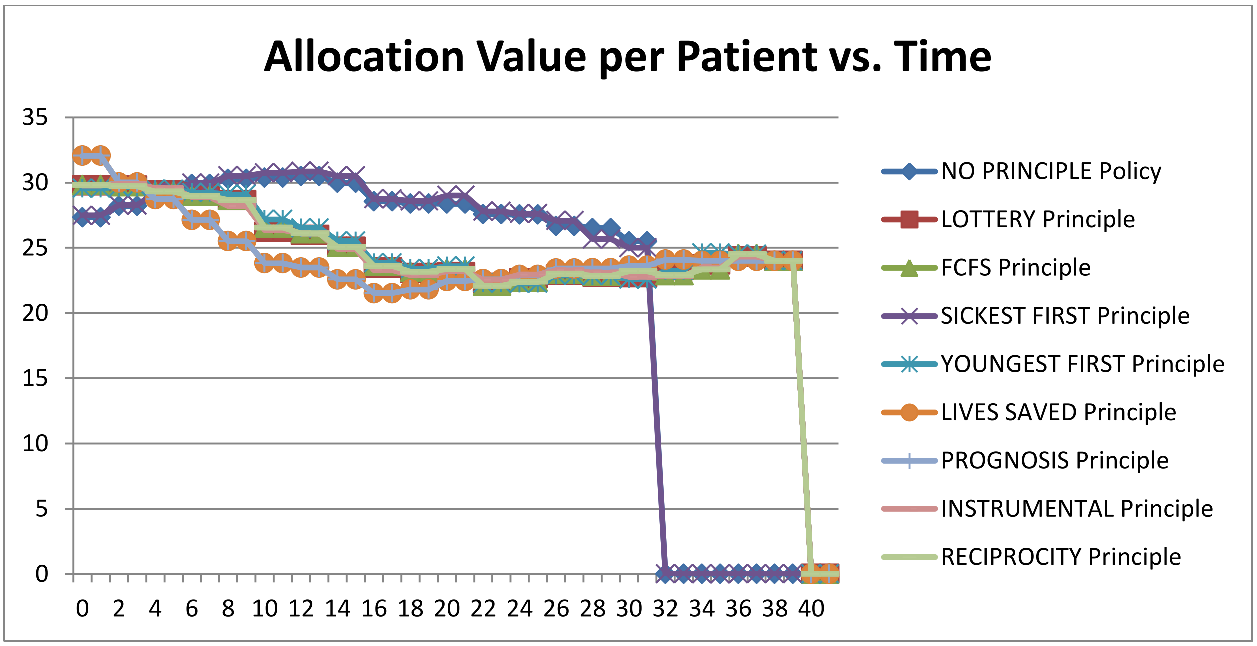

Figure VH-14a – Hospital: Allocation Value per Patient over Time (Sensing=2)

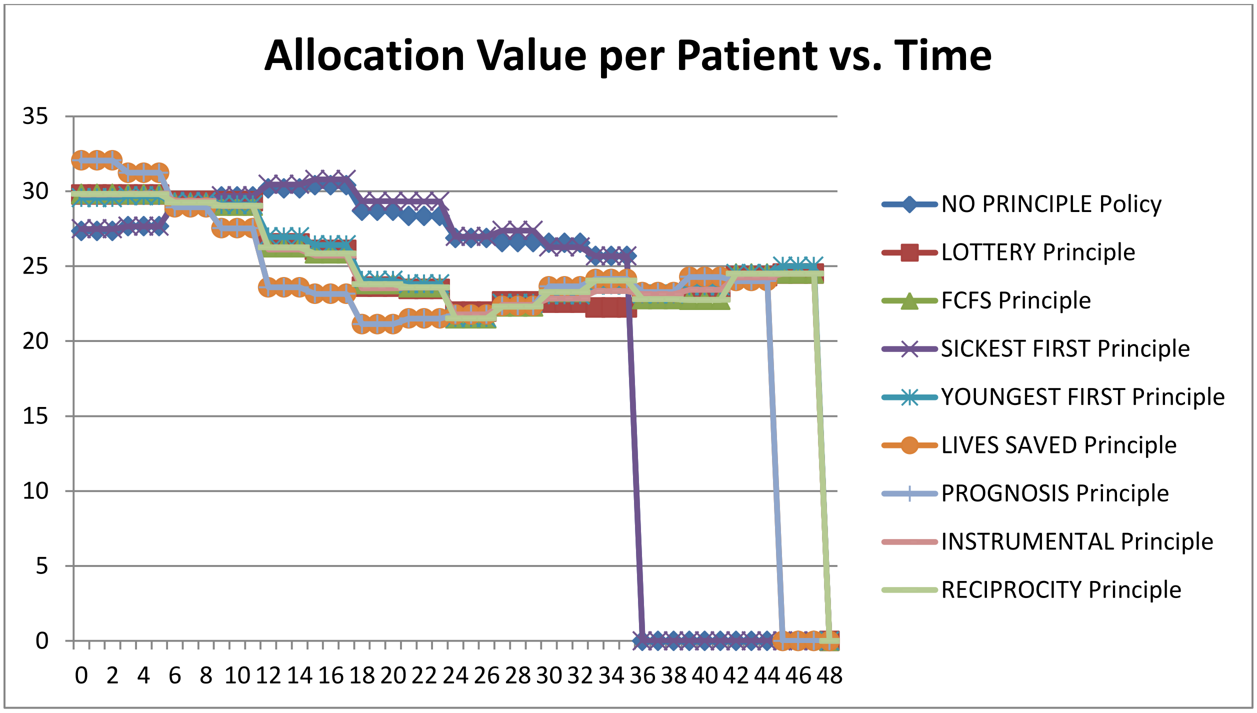

Figure VH-14b – Hospital: Allocation Value per Patient over Time (Sensing=3)

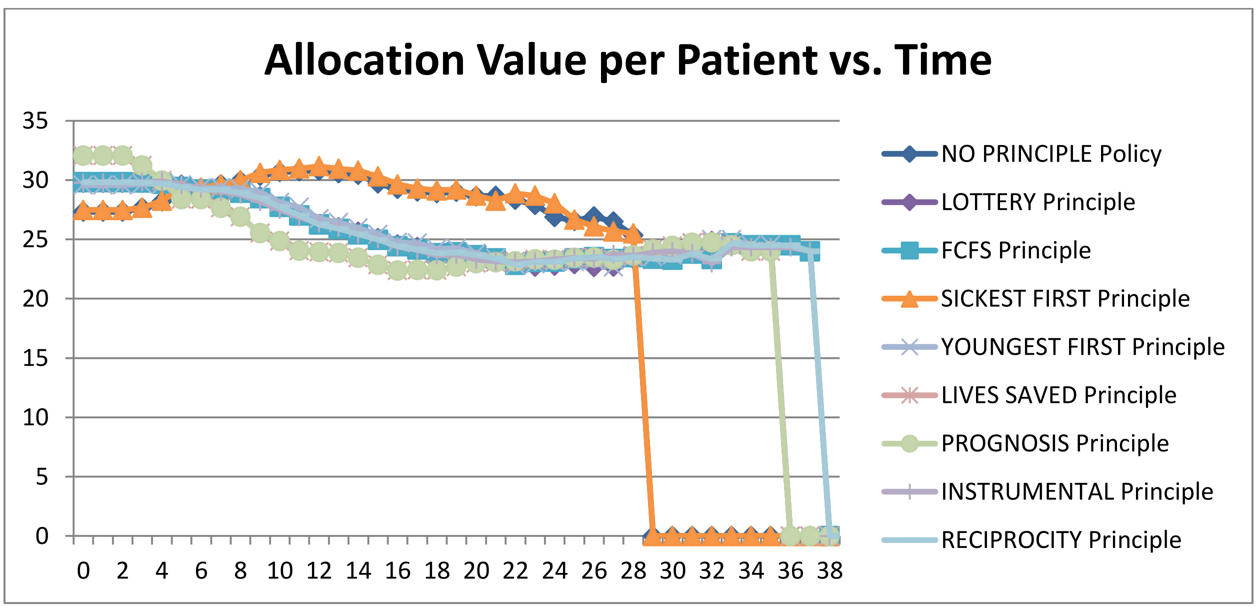

Figure VH-18a – Hospital: Allocation Value per Patient over Time (Sensing=1, Buffer=1)

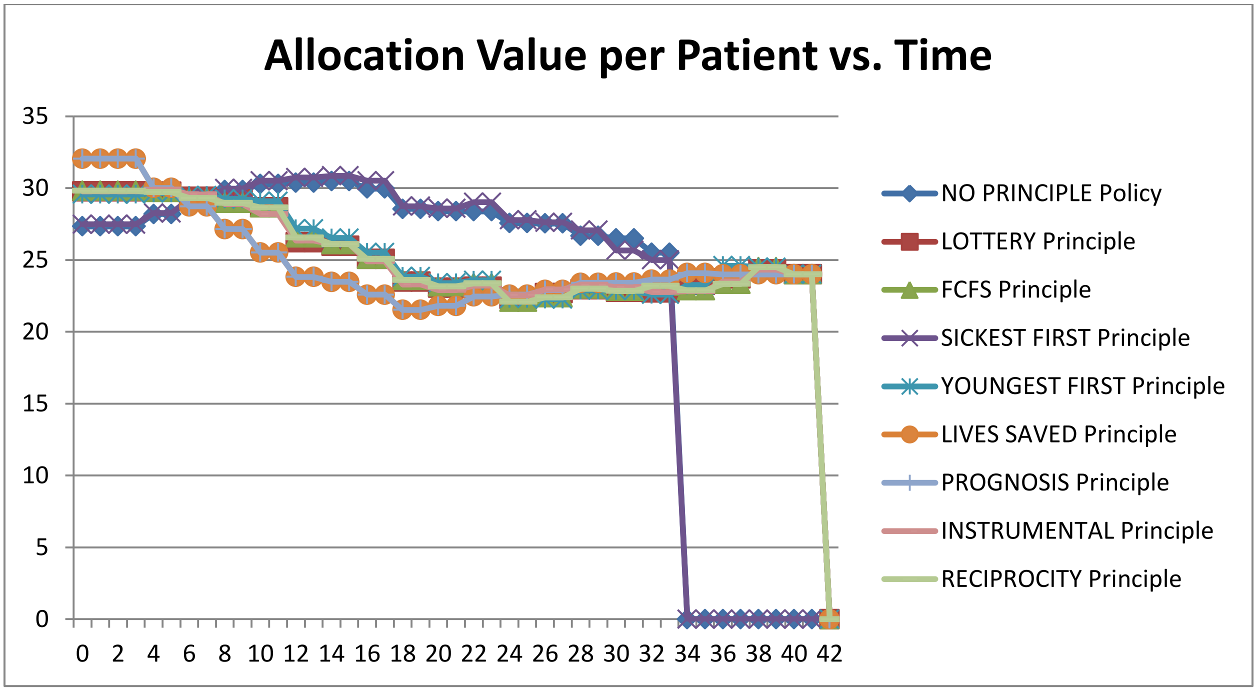

Figure VH-18b – Hospital: Allocation Value per Patient over Time (Sensing=2, Buffer=1)

5.2.3 Additional Output of Value

Appendix D shows output from a Local Search for Ambulance and Hospital Stages. It illustrates the kind of output that can be tracked by a system designer or presented to a decision maker. In so doing, we can view how each new neighbor that is being considered in the local search appears, along with how the allocation values change, to result in appropriate swaps or changes. Appendix F in turn shows some of the attribute values that can be displayed when presenting the results. In Decision Support mode, a decision maker may propose a particular user preference. When final proposed allocation values are determined, the user may view the relative difference between what our algorithms suggest and what the user was considering. Appendix C illustrates the kind of output that can be generated at this stage. In order to view the effect of different stopping criteria (number of random restarts) some sample output is offered in Appendix E. This would be useful to display to a system designer.

5.3 Summary

We have devoted considerable discussion to the usefulness of the implementation for decision makers. We point out that these users in the domain of application of healthcare are ones who will have to ultimately make the final decisions, themselves, and as such a general impression of the value of our methods will be most important to them. A visualization of how patients categorized by severity are allocated is especially valuable to help a user understand how the resources are allocated. The policy concept allows for users to view particular combinations of preferences and their relative differences. We return to discuss some avenues for future work with the experimentation within Section 6.3.

The second primary usage of the graphs displayed in this chapter is to demonstrate the value of our local search approach. There are several graphs were comparisons between policy based approaches and principle based approaches are apparent. The relative value of the policy approach compared to using principles of allocation is best viewed by examining the output from using no-principle, compared to any of the selected principles. We observe that how the weights of different factors were set in our cost functions might be an influence. If, for instance, we set a higher weight for the PrincipleCostPenalty, the differences might be more dramatic. We discuss this further in Section 6.3.

While the sickest first principle often produced effective performance our policy based approach consistently offered at least equal or better performance as well. In Section 4.3.1.1, we shed some light on why improvements should arise with the use of local search. In this chapter, we have also discussed the value of examining graphs that display what happens to patients at different severity levels together with graphs which track total cost (allocation value).

Chapter 6
Conclusion, Discussion and Future Work

In this chapter, we summarize some of the key accomplishments of this thesis, discuss several valuable directions for future research, and provide final reflection on the value of our work in comparison with competing approaches.

6.1 Contributions

This thesis outlines a particular solution for multiagent resource allocation for time critical dynamic environments that enables real-time decision making. We begin by listing some of the individual contributions of this research.

Multiagent resource allocation

- Outlined in full a solution based on local search which is capable of
    - Coping with time-critical, real-time decisions
        - Demonstrated through effective times for completing allocations
        - Achieved in part through not using continuous values – i.e. categorizing different attributes (e.g. waiting times in terms of being between in a range of possible values) helped to speed up search
    - Coping with dynamic environments
        - Through solutions developed for sensing and updating (to go beyond the current state)
    - Integrating effective solutions for swaps and restarts in the search
        - Opting for a first improvement approach to find neighbors (i.e. not going through all possible neighbors) to speed up processing and to allow more random restarts, avoiding local minima
    - Empirically determining the best number of restarts to incorporate (i.e. used 100 restarts since the cost of doing so in time was minimal)
- Mapped out how to integrate local search and social welfare consideration

    - Examined a number of distinct social welfare functions
    - Outlined how to calculate global costs/utilities sensitive to social welfare to drive local search decisions

Decision Making Support for Resource Allocation

- Integrated into our proposal for resource allocation a mode of operation where users can acquire quantifiable feedback about alternative choices for the allocation decisions (e.g. social welfare preferences)
- Allowed for various metrics to be displayed for decision support

Parameter Values for Modeling the Allocation Problems

- Proposed an integration of sensing for updating parameter values

Validation of the Proposed Resource Allocation Framework

- Detailed simulations for both Allocation and Decision Making modes
- Decisions for how to model and map out values, what to compare, etc.

We also present below a series of contributions for the specific healthcare application of handling mass casualty incidents, which has been the focus of our attention throughout this thesis.

- Modeled a mass casualty incident
    - Able to explore resource allocation in the emergency department environment when the resources are limited
    - Solved by local search to determine the best allocation according to the policy
    - Solutions took very little time, which is essential for emergency departments since they are time critical
    - Explored the Ambulance and Hospital stages
- Modeled the dynamic nature of the emergency department
    - Allocation are updated according to the current situation
    - Resources will be reallocated as the situation changes

    - Simulations of the Ambulance stage until every victim reached the hospital and the Hospital stage until every patient has been discharged from the hospital
- Social welfare functions and principles of allocation offered many approaches to allocate the required resources to the patients
    - Provides comparisons and reasoning for allocations
    - Useful for validation purposes
    - Valuable for hospital administrators to evaluate the approaches and simulate situations in order to analyze the results
- Decision support for the medical staff members to help make their allocation decisions
    - Confirms that the allocation is best or provides a better alternative as a comparison
    - Medical staff member make these decisions in stressful situation and in a short amount of time so the alternative solution could be a decision that they would have made if they had more time
- Provided the building blocks to create a corrector system for ambulances and hospitals which maps out how to engage the framework outlined in this thesis within a real environment, with actual medical professionals, patients, ambulances.

6.2 Discussion and Related Work

In this section, we return to clarify how our proposed approach for MARA is well suited for applications that have certain characteristics, as listed in Chapter 1. We then move on to compare our approach with that of other researchers.

The application that we chose to be the centerpiece of this thesis is that of healthcare, examining both an Ambulance and Hospital stage. We clarify here how this particular application has all the characteristics that we described as desirable for our particular investigation of multiagent resource allocation. The resources that we model are indivisible (i.e. patients will not be given a fraction of an ambulance or a doctor). Some of our resources are shareable (e.g. the same hospital can be used as the destination of two different ambulances) whereas others are not shareable (e.g. each ambulance transports only one victim). The multiagent system is cooperative: we have discussed hospital administrators as one possible decision making group, and view the needs of all parties in the environment as being provided transparently and honestly. Agent preferences, for us, are dictated by their current state; their needs for resources are set by their location, their severity and their injuries. Differing patients can have different resource requirements when reaching the hospital, still. As illustrated in our example in Section 4.3.1.1, each patient prefers a resource that will provide the best possible care, still. We have indeed settled on a goal of minimizing the cost of allocating resources in a poor fashion. As illustrated in the setting used for our implementation and discussed in the text, some victims or patients may be without a resource at any given point in time. Dynamic adjustments to allocations are supported, and a refresh of parameter values occurs with some frequency.

In Chapter 2, we discussed alternative models for patient scheduling, drawn from artificial intelligence literature. Compared to other approaches for handling patients in healthcare scenarios we develop a framework Ambulance and Hospital stages at the same time. Other research more typically handles the Hospital stage alone [13, 21, 31, 33, 43]. What is also distinct about our approaches is the fact that most of the competing models do not consider the influence of principles of allocation and thus do not guide the solution based on the decision maker’s preference with respect to such principles.

In Chapter 4, we discussed our aim of trying to effectively model the Ambulance and Hospital stages where MARA concerns arise and presented our proposed parameters, parameter values and cost functions. The challenge of effectively modeling healthcare scenarios for artificial intelligence systems is one that was also faced by Cohen et al. [9]. That work was concerned with reasoning about when to interact with doctors in order to tend to patients and included a focus on properly modeling bother and bother cost. The doctors and patients also needed to be modeled and in this case the options of high, medium or low skill, and high, medium or low severity were chosen. Our proposed user modeling offers a richer set of options here and also includes additional modeling characteristics in terms of procedures and resource capabilities.

6.2.1 Contrast with Other MARA Approaches

Our approach to MARA can be characterized as centralized, with provision for domain-specific cost functions, and employing local search for solutions, minimizing total cost according to social welfare.

VC Auction [20] research is somewhat relevant to ours. In these auctions, if the parties have sufficient money, they can submit bids which determine who gets which resource, along with a paid cost which compensates who fails to get a resource. This is provably optimal: true valuations for bids are elicited and utilitarian social welfare can be maximized. The solution is NP hard to compute, so approximations are needed. Since our problems are small, we may be able to perform the computations precisely but there are primary differences from the applications we are most interested in. For one, the concept of paying money makes less sense to introduce in our eminently cooperative environment.

Paulussen et al.'s work [33] is relevant to ours in that it comments on resource allocation in healthcare settings. This work is focused on preemption and in particular on addressing possible cyclical requests for resources, considering opportunity costs. We allow some swapping of resources when first trying to settle on the proposed allocation but then consider that after a certain time buffer, the resources allocated are fixed. We do this to provide a more simplified view, so that we can focus instead on investigating the relative value of our local search approach compared to solutions directed according to principles of allocation.

As mentioned in Chapter 2, the work of Doucette is also relevant in considering healthcare applications for MARA [14, 15]. In that work, the cost function is based on a single swap and asking whether this will improve the solution. Our cost function has additional components that are considered. As it is centralized, there is full information.

A number of artificial intelligence researchers have explored how to address dynamically changing environments, for resource allocation. This is the central consideration of the work of Pinhey, Doucette, and Cohen [35], which is inspired by the work of Doucette [14, 15] to also account for dynamic task arrivals. In contrast with our work, this research adopts a decentralized approach and is focused on reasoning about which parties are best to assume a current task, taking into account the bother that will be generated. A major focus of that research is also on how best to coordinate requests for resources using proxy agents, allowing preemptions when it is expected that agents who have lost resources can still complete acceptable, alternate plans. The way that dynamic arrivals are handled is to re-evaluate resource needs with each new time tick. In our work, we reason about how to set our initial time buffer, before locking in resources and continuing with that particular allocation.

Other research investigating dynamic resource allocation is also focused on a decentralized, distributed solution. Macarthur et al.'s [26] paper focuses on task allocation where agents can form coalitions to finish a task. Finding the optimal number of agents to form a coalition is different from our approach, since we assume that a bundle of requested resources is assigned. Chapman et al.'s [7] paper takes a game-theoretic approach to dynamic task allocation. This is in contrast to our work, which is more cooperative and where the assembly of teams is simply a part of the bundle requests that are defined.

The best way to view our particular stance on multiagent resource allocation is that we seek to clarify how local search can be used to determine the allocations, when social welfare functions are employed to drive the utility calculations and where minimizing costs is desired. We adopt this particular view of MARA in order to then facilitate the provision of output that may assist decision makers in environments such as healthcare. Not only do we run algorithms to present proposed final allocations, but also we outline how various competing preferences can be input, to show comparisons. Introducing the concept of a cost function and the modeling of total cost enables us to drill down to various domain-specific important considerations, to include them in our modeling and thus to have them influence the allocations that are proposed.

6.3 Future Work

There are several valuable avenues for future research which may extend our current model and its application to existing topic areas.

6.3.1 Hyper-heuristics for Policy Specification

We mention the possibility of introducing hyper-heuristics [6] as a way of offloading the requirement of specifying a policy as part of the input to decision making. For future work, one direction could be to use hyper-heuristics in order to select the most appropriate policy. Since allocation values change considerably based on the social welfare function, we would likely ask the decision maker to continue to specify the social welfare function; hyper-heuristics would help in identifying the preferred principle of allocation. Another use of hyper-heuristics is for heuristic generation. If we were to employ this direction, this could assist in setting the weights of the assignment values as part of the total cost evaluation function.

6.3.2 Modeling Other Dynamic Parameter Values

Earlier we suggested that our modeling of the environment could be extended to include items like roadblocks that affect ambulance travel. For future work, we could extend our current proposed modeling for both ambulance and hospital scenarios to consider a more extensive set of parameters which might influence our allocation decisions.

6.3.3 Extensions to Sensing

There are various ways in which we could extend our current proposal for sensing. One issue is dealing with noise in the sensor readings. Another is to continue to learn about when it is best to do the sensing or where the sensors should be placed. We could benefit from ongoing research such as [36] that is already focused on effective placing of sensors in hospital settings.

6.3.4 Other Domains

Other domains for multiagent resource allocation would benefit from our approach. Two domains, for example, include firefighting [41] and natural disasters. These domains would be time-critical and require real-time decisions.

A Firefighter application describes a few fires occur throughout an area (forests, homes, etc.). The areas on fire require resources to stop the spread and to extinguish the fire. These resources include fire trucks, firefighters, and equipment. The cost for that area will increase as the fire increases. The cost of the area will also be determined by the distance and priority of the area. Stages include initial fire alarm, new fires discovered, and changes in fire.

A Natural Disaster application describes resources that are needed by the victims of a natural disaster. The victims are in groups of various sizes (stuck in their homes, offices, schools, etc.). Resources include food, water, first aid, and clothes. The cost of each group will be determined by the resources they want and were allocated. Stages include initial disaster, new disaster, and changes to the group.

If we are exploring other domains as future work we would need to ensure that our algorithm is sufficiently general in order for it to operate in these domains. We would also want to run simulations to confirm the value of our approach. In different domains, decision makers may want to see additional information when displaying the graphs. We may also need to either extend or adjust the set of principles of allocation that we used. For example, sickest first may need to be construed to “in most danger” for the above two domains.

6.3.5 Other Experiments

We could run additional simulations as part of our validation. For example, we could experiment with a much larger number of victims, in scenarios with very limited or extensive available resources. In might be interesting to also consider trying to handle new patients who have arrived, independent of the mass casualty incident. We could also run experiments to address overcapacity by allowing less critical patients to move to different hospitals. Another scenario to explore further is where there are actually sufficient resources for patients in the hospital. Finally we could code up other social welfare functions (e.g. Rank Dictator, Elitist) in order to analyze the value of these options.

6.3.5.1 Alternate weights and parameter values

Different weights for cost function components would be instructive to explore. This may lead to more dramatic comparisons between the two competing approaches. For example, assigning a larger penalty for the PrincipleCostPenalty parameter could reflect a greater significance for this component and would adjust the plots. Experimenting with different profiles of parameter values (e.g. scenarios with a large proportion of severity 1 patients) would be instructive.

6.3.6 Analysis of Experimental Conditions

In our validation various parameter values were set randomly. For future work it would be useful to determine whether certain specific values end up contributing significantly to the results that are obtained. Mitchell et al. [28] have observed that testing with randomly generated values at times presents a challenging distribution of instances, in other words one that would make it difficult for the local search to successfully complete in a relatively short period of time. It would be useful to conduct analysis to identify these cases.

6.3.7 Cost Functions

Another new direction would be to model and reason with different kinds of cost. For example, the cost of bothering doctors is useful to consider as seen in Cohen et al.'s work [9]. This would assist in modeling overutilization of resources in order to improve our own performance.

6.3.8 Preemption

At the moment we allow various swaps and changes for our resource allocation. Researchers such as Doucette [15] have examined the issue of preemption more extensively. Considering more sophisticated preemption and how to measure its cost might be a valuable path for future work.

6.3.9 Buffer Times

It may be possible to model the amount of time where the resources would still be able to be reallocated more effectively, by using learning to acquire a better understanding of how the emergency departments in hospitals currently operate [47]. We could then experiment with the suggested values for the time buffering and analyze results in order to determine how best to set the time buffers.

6.3.10 Expanding Decision Support

For future research it would be valuable to explore what kind of user interface would be best for the decision support offered to our user. If we were to construct a user interface, this would then allow us to conduct user studies of our corrector system.

6.3.11 Robust Allocations

With the framework that we have developed, it would be valuable to begin to learn how to avoid brittle allocations, to try to make them more robust. Robust allocations would ensure minimal changes when reallocating new resources. This provides for less instability in the allocation, improving the flow and thus the efficiency of the allocation process. It would also be valuable to identify those circumstances that make allocations more brittle in order to avoid them.

6.3.12 Multiagent Planning for Resource Allocation

We have done an initial exploration of a multiagent planning approach to resource allocation. In this scenario each agent has a plan that is merged together and some individuals may have their plans altered. In contrast with the model used by Cox and Durfee [10], we would propose to integrate the use of social welfare functions and principles of allocation. This would suggest an entirely new direction for resolving resource allocation, which could be compared to our current approach.

6.3.13 MDPs and Other Machine Learning Approaches

Another entirely new path would be to consider using MDPs, evolutionary algorithms, or other machine learning approaches in place of the local search used in our proposal. We could for instance try to compare the results from our framework with those from these alternative designs. This could include comparisons of timing, accuracy, and memory requirements.

6.3.14 Alternative Local Search Methods

Future work can investigate other search paradigms such as tabu search or simulated annealing in order to compare these with our hill climbing first-improvement approach. We would compare the different running times and effectiveness of these approaches. We expect to discover various tradeoffs between time and accuracy improvements. Another direction would to use a SAT solver [3, 22] to determine the solution in place of local search. One concern would be the time required, because in domains such as healthcare quick responses are needed.

6.4 Closing Remarks

In all, we have presented a framework for multiagent resource allocation in dynamic, time-critical environments that is helpful for applications such as healthcare. We have demonstrated that a local search approach for CSPs can work well, through a series of simulations, and have also clarified where decision support can be provided. Compared to other researchers examining the problem of patient scheduling, we focus on a centralized approach and in the handling of emergency scenarios, as well incorporating social welfare functions; we also contrast with designs that are based on machine learning. In tune with current research on decision support for medical applications, we allow for varying inputs and ensure that humans are making the final decisions. We have also outlined a number of opportunities for future research in order to see the ongoing potential of our proposed approach.

Bibliography

1. American Hospital Association. (2012) HIPAA Updated Guidelines for Releasing Information on the Condition of Patients [Online] Available: http://www.aha.org/advocacy-issues/tools-resources/advisory/96-06/030201-media-adv.shtml .
2. E. M. Atkins, T. F. Abdelzaher, K. G. Shin and E. H. Durfee, "Planning and resource allocation for hard real-time, fault-tolerant plan execution," Autonomous Agents and Multi-Agent Systems, vol. 4, no. 1-2, pp. 57-78, 2001.
3. M. Bader-El-Den and R. Poli, "Generating SAT local-search heuristics using a GP hyper-heuristic framework," in Artificial evolution, 2008.
4. C. Bessiere, "Arc-Consistency in Dynamic Constraint Satisfaction Problems," in AAAI, 1991.
5. C. C. Branas, R. F. Sing and A. D. Perron, "A case series analysis of mass casualty incidents," Prehospital Emergency Care, vol. 4, no. 4, pp. 299-304, 2000.
6. E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan and J. R. Woodward, "A classification of hyper-heuristic approaches," in Handbook of Metaheuristics, Springer, 2010, pp. 449-468.
7. A. C. Chapman, R. A. Micillo, R. Kota and N. R. Jennings, "Decentralised dynamic task allocation: a practical game: theoretic approach," in Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems- Volume 2, 2009.
8. Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaitre, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar and P. Sousa, "Issues in multiagent resource allocation," Informatica (Slovenia), vol. 30, no. 1, pp. 3-31, 2006.
9. R. Cohen, H. Jung, M. W. Fleming and M. Y. Cheng, "A user modeling approach for reasoning about interaction sensitive to bother and its application to hospital decision scenarios," User Modeling and User-Adapted Interaction, vol. 21, no. 4-5, pp. 441-484, 2011.
10. J. Cox and E. Durfee, "Efficient and distributable methods for solving the multiagent plan coordination problem," Multiagent and Grid Systems, vol. 5, no. 4, pp. 373-408, 2009.
11. M. de Weerdt and B. Clement, "Introduction to planning in multiagent systems," Multiagent and Grid Systems, vol. 5, no. 4, pp. 345-355, 2009.
12. R. Dechter and A. Dechter, Belief maintenance in dynamic constraint networks, University of California, Computer Science Department, 1988.
13. K. Decker and J. Li, "Coordinated hospital patient scheduling," in Multi Agent Systems, 1998. Proceedings. International Conference on, 1998.
14. J. A. Doucette and R. Cohen, "Addressing Preemption Costs in Multi-Agent Resource Allocation for Medical Applications," in Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
15. J. A. E. Doucette, "An Ex-Ante Rational Distributed Resource Allocation System using Transfer of Control Strategies for Preemption with Applications to Emergency Medicine," 2012.
16. S. Einav, Z. Feigenberg, C. Weissman, D. Zaichik, G. Caspi, D. Kotler and H. R. Freund, "Evacuation priorities in mass casualty terror-related events: implications for contingency planning," Annals of surgery, vol. 239, no. 3, p. 304, 2004.
17. D. Feeny, W. Furlong, G. W. Torrance, C. H. Goldsmith, Z. Zhu, S. DePauw, M. Denton and M. Boyle, "Multiattribute and single-attribute utility functions for the health utilities index mark 3 system," Medical care, vol. 40, no. 2, pp. 113-128, 2002.

18. V. Heimly and O. Nytro, "Clinical guidelines as decision support for referrals from primary care," in Information Reuse and Integration (IRI), 2011 IEEE International Conference on, 2011.
19. H. Jung and R. Cohen, "A model for reasoning about interaction with users in dynamic, time critical environments for the application of hospital decision making," in Advances in Artificial Intelligence, Springer, 2010, pp. 319-323.
20. J. Kalagnanam and D. C. Parkes, "Auctions, bidding and exchange design," In Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era, pp. 143-212, 2004.
21. E. Kanaga, P. Darius and M. Valarmathi, "An efficient multi-agent patient scheduling using market based coordination mechanism," in Intelligent Agent \& Multi-Agent Systems, 2009. IAMA 2009. International Conference on, 2009.
22. H. Kautz and B. Selman, "The state of SAT," Discrete Applied Mathematics, vol. 155, no. 12, pp. 1514-1524, 2007.
23. M. Li, C. Fung, K. Sampigethaya, R. Robinson, R. Poovendran, R. Falk, F. Kohlmayer and A. Koepf, "Public key based authentication for secure integration of sensor data and RFID," in Proceedings of the 1st ACM international workshop on Heterogeneous sensor and actor networks, 2008.
24. L. Liao, D. Fox and H. Kautz, "Extracting places and activities from gps traces using hierarchical conditional random fields," The International Journal of Robotics Research, vol. 26, no. 1, pp. 119-134, 2007.
25. C. M. Lilly and E. J. Thomas, "Tele-ICU: experience to date," Journal of Intensive Care Medicine, vol. 25, no. 1, pp. 16-22, 2010.
26. K. S. Macarthur, R. Stranders, S. D. Ramchurn and N. R. Jennings, "A Distributed Anytime Algorithm for Dynamic Task Allocation in Multi-Agent Systems," in AAAI, 2011.
27. N. Meuleau, M. Hauskrecht, K.-E. Kim, L. Peshkin, L. P. Kaelbling, T. L. Dean and C. Boutilier, "Solving very large weakly coupled Markov decision processes," in AAAI/IAAI, 1998.
28. D. Mitchell, B. Selman and H. Levesque, "Hard and easy distributions of SAT problems," in AAAI, 1992.
29. S. Mittal and B. Falkenhainer, "Dynamic constraint satisfaction," in Proceedings Eighth National Conference on Artificial Intelligence, 1990.
30. N.-T. Nguyen, T. T. Nguyen, M. Roos and J. Rothe, "Complexity and approximability of social welfare optimization in multiagent resource allocation," in Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3, 2012.
31. C. Niemann and T. Eymann, "Beyond Automation: Hybrid Human-Computer Decision Support Systems in Hospitals," in Engineering of Autonomic and Autonomous Systems, 2008. EASE 2008. Fifth IEEE Workshop on, 2008.
32. L. P. Østerdal, "Axioms for health care resource allocation," Journal of Health Economics, vol. 24, no. 4, pp. 679-702, 2005.
33. T. Paulussen, N. Jennings, K. Decker and A. Heinzl, "Distributed patient scheduling in hospitals," in Proceedings of the 18th international joint conference on Artificial intelligence, 2003.
34. G. Persad, A. Wertheimer and E. Emanuel, "Principles for allocation of scarce medical interventions," The Lancet, vol. 373, no. 9661, pp. 423-431, 2009.
35. G. Pinhey, J. Doucette and R. Cohen, "Distributed Multiagent Resource Allocation with Adaptive Preemption for Dynamic Tasks," in Proceedings of The 13th International Conference on Autonomous Agents and Multiagent Systems, 2014.

36. I. Popovici, "Patient-centered Perspectives of Communication and Handover between the Emergency Department and General Internal Medicine," 2011.
37. G. Priya, P. Anandhakumar and K. Maheswari, "Dynamic scheduler-a pervasive healthcare system in smart hospitals using RFID," in Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on, 2008.
38. F. Rossi, P. Van Beek and T. Walsh, Handbook of constraint programming, Access Online via Elsevier, 2006.
39. D. Shaft and R. Cohen, "A Multiagent Approach to Ambulance Allocation Based on Social Welfare and Local Search," in Machine Learning and Applications (ICMLA), 2013 12th International Conference on, Workshop: Bioinformatics and Health Informatics, 2013.
40. M. Taboada, E. Cabrera, E. Luque, F. Epelde and M. Iglesias, "A decision support system for hospital emergency departments designed using agent-based modeling and simulation," in Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on, 2012.
41. A. Tsang, K. Larson and R. McAlpine, "Resource Sharing for Control of Wildland Fires," in Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
42. G. Verfaillie and T. Schiex, "Solution reuse in dynamic constraint satisfaction problems," in AAAI, 1994.
43. I. Vermeulen, S. Bohte, K. Somefun and H. La Poutré, "Multi-agent Pareto appointment exchanging in hospital patient scheduling," Service Oriented Computing and Applications, vol. 1, no. 3, pp. 185-196, 2007.
44. A. Weale, "From equity to equality? A comment on the contribution by Alastair Campbell," HEALTH CARE PRIORITY SETTING, p. 40, 2003.
45. G. Weiss, Multiagent systems: a modern approach to distributed artificial intelligence, The MIT press, 1999.
46. S. Weng, T. Wu, G. Mackulak and J. Fowler, "Distributed resource allocation for healthcare systems," in Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on, 2008.
47. H. J. Wong, D. Morra, M. Caesar, M. W. Carter and H. Abrams, "Understanding hospital and emergency department congestion: an examination of inpatient admission trends and bed resources," CJEM, vol. 12, no. 1, pp. 18-26, 2010.
48. W. Wong, D. O'Hare and P. Sallis, "A goal-oriented approach for designing decision support displays in dynamic environments," in Computer-Human Interaction, 1996. Proceedings, Sixth Australian Conference on, 1996.
49. J. Xiao, L. Osterweil and Q. Wang, "Dynamic scheduling of emergency department resources," in Proceedings of the 1st ACM International Health Informatics Symposium, 2010.

Appendix A
Sufficient Ambulance Details and Results

We employed a cost function for this initial exploration where high costs were preferred. In this Appendix, we provide details on how we modelled and implemented the Sufficient Ambulance stage. In particular we present our proposed cost functions and then display various results from implementation.

A.1 Cost Functions

In Algorithm A.1, we see the total cost function that we would like to optimize. (This is a counter-intuitive use of the word cost; a high value represents a favorable outcome). There are two components to calculate the total cost in the Sufficient Ambulance stage. The first is a cost arising from the hospitals namely a penalty for being overcapacity. For the penalty of being over capacity, we will be subtracting a CAPACITYPENALTY (set to -15 in our implementation) for each patient over capacity in a hospital. The second is a cost arising from the individual allocations for each of the victims (SingleSolution costs) aggregated according to the specific social welfare function selected as part of the input. This cost is referred to as calculateCost function, which depends on the calculateCostDrive and calculateCostResource functions.

Algorithm A.1: Calculating the total cost

```
Input: Each assignment of patients to hospitals (singleSolution), Hospitals, Social welfare function
Output: Final cost based on Utilitarian or Egalitarian social welfare
Initialize totalCost = 0;
for each singleSolution
  cost = cost of the singleSolution;
  Combine cost to the totalCost according to the social welfare function;
end
for each hospital in hospitals
  cost = 0;
  if hospital is over capacity then
   cost = number of patients over capacity * CAPACITYPENALTY;
  end
  Add cost to the totalCost;
end
return totalCost;
```

$$calculateCost = calculateCostDrive + calculateCostResource$$

The calculateCostDrive function is shown below. This cost will depend on the distance to the hospital as well as the victim's severity, initial time, and the ambulance driver. To summarize the function, if the victim's severity is critical and the drive is not close then value is given by the EMS driver skill plus initial time subtracted by HIGHCOST (15 in our implementation). If the victim's severity is serious and the drive is far then value is given by the EMS driver skill plus initial time subtracted by MEDIUMCOST (which we set to 10). If the victim's severity is fair, the drive is far, and the driver is a rookie then value is given by the EMS driver skill plus initial time subtracted by LOWCOST (which we set to 5). Otherwise, the value of the severity is multiplied by the distance to the hospital then adding the values for the ambulance driver and the initial time before the ambulance left.

The calculateCostResource function will return better results if the victim’s resource needs are met. If the hospital’s resource is greater than or equal to the victim needs then the overall cost is the difference between the values of the resources. However, if the hospital’s resource is worse than the victim needs then there will be a penalty, which is called NEGPENALTY (set to -20 in our implementation).

$$calculateCostDrive = \begin{cases} vD + vIT - HIGHCOST, & vSev = CRITICAL\_SEVERITY \wedge hdist \neq DISTANCE\_CLOSE \\ vD + vIT - MEDIUMCOST, & vSev = SERIOUS\_SEVERITY \wedge hdist = DISTANCE\_FAR \\ vD + vIT - LOWCOST, & vSev = FAIR\_SEVERITY \wedge hdist = DISTANCE\_FAR \wedge vD = DRIVER\_ROOKIE \\ (hdist * vSev) + vD + vIT, & Otherwise \end{cases}$$

$$calculateCostResource = \begin{cases} hRes - vRes, & resource\ at\ hospital\ is\ better\ than\ or\ equal\ to\ victim's\ request \\ NEGPENALTY, & resource\ at\ hospital\ is\ worse\ than\ victim's\ request \end{cases}$$

A.2 Modeling and Implementation

To implement this problem, we decided to program in the Java programming language. Figure 4.1 in Section 4.1.3 includes the important sections of a class diagram for this problem. These will serve to clarify the proposed modeling for the Sufficient Ambulance stage.

The Victim class represents the injured individual in the mass casualty incident. Four integers are used to describe a victim. Resources represent the resources needed for a victim, which will be high, medium, or low. The lower the resources needed, the higher the value. Severity represents the condition of the victim, which will be good, fair, serious, or critical. The lower severity of the victim causes a higher value. Driver represents the skill of the EMS driver, which is either a skilled veteran or a rookie (e.g. in our implementation, skilled has a value of 2 and rookie has a value of 1). Finally, initialTime represents the time from the beginning of the incident until the victim left in the ambulance. We decided to model initialTime as one of the set of possible time ranges (e.g. in our implementation we use less than ten minutes, between ten and fifteen minutes, between fifteen to twenty minutes, between twenty to thirty minutes, or greater than thirty minutes); the usage of these discrete values assist in performing the local search efficiently. The less time initially, the better the value.

The Hospital class represents a hospital in the domain of the problem. Four integers are used to describe a hospital. Resources represent the resources available for a hospital, which will be high, medium, or low. The more resources available at the hospital cause the higher value. Capacity represents the amount of patients the hospital can accept before becoming full. A hospital can be over the capacity, which is similar to the real world. The assumption for capacity is to ignore the patients already in the hospital and just present the number of patients that the hospital can still accept. There are three possible values to set for the capacity. The first is a high capacity (which in our implementation represents 15 new patients). The next is a medium capacity (which represents 12 new patients). The final is a low capacity (which represents 8 new patients). It is better to have a larger amount for new patients. Distance represents the distance from the mass casualty incident to the hospital, which will be close, medium, or far. The closer the distance from the incident to the hospital causes the higher value. Finally, currentPatients represents the current patients at the hospital. Having the current number of patients be less than or equal to the capacity is beneficial. This is determined by using the function diffCapacity(), which returns capacity subtracted by the current number of patients in that hospital. Whenever a patient is added or removed from a hospital the addPatient() and removePatient() functions are used to change the currentPatients value.

As mentioned in Section 4.1.3, the SingleSolution class is the pairing of a victim with a hospital and the Solution class represents all of the assignments of victims to hospitals. Details for determining the costs are mentioned in Section A.1. Moreover, details about the local search for the Sufficient Ambulance stage are discussed in Section 4.1.3.

The Problem class is the main class. This is where hospitals and victims are initialized using initHospitals() and initVictims(). As stated previously, the number of patients is 30 and the number of hospitals is 3 for most of experimental results. A random number generator is used to determine the values for the victims and hospitals based on their possible values. Due to the random generator, some of the soft constraints may never be satisfied. To have consistent results between the local search strategies, a seed is used for the random number generator.

A.3 Experimental Results

The experimental results were produced by running the local search on 100 different generated scenarios, where each scenario had 30 victims and 3 hospitals. Given that the random generator used a seed, the same 100 scenarios were run for each approach used. There are 6 approaches based on the social welfare and initial state. There are two options for the social welfare, which are Utilitarian and Egalitarian. There are three options for the initial state, which are even, single, and random.

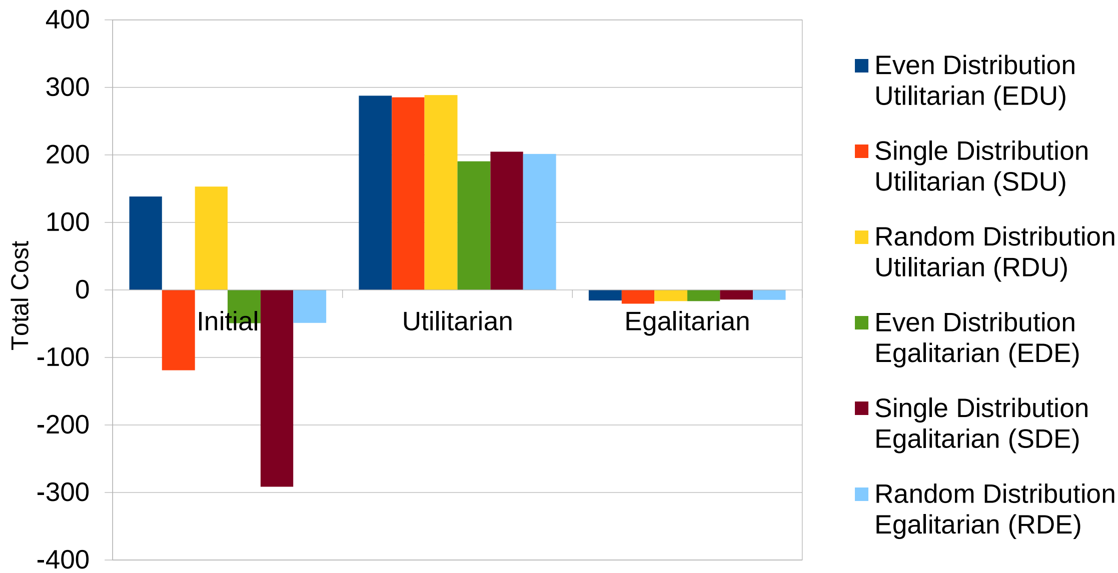

Figure A.1 – Total cost graph

Figure A.1 shows the total cost based on the approach take. As explained, the cost represents the value of the victims receiving the resources (recall that a high cost is desirable). The graph shows the total cost of the initial state, the social welfare used in the approach, and the social welfare of the other approach. In Figure A.1, the first three bars in each group used the Utilitarian approach and the last three bars in each group used the Egalitarian approach, to drive the calculations in Algorithm 4.1 (in Section 4.1.3), resulting in the proposed final allocation. To further expand on this, the first three bars in the initial section show the initial value calculated with the Utilitarian social welfare function, the first three bars in the Utilitarian section show the final value calculated with the Utilitarian social welfare function (i.e. the sum of the cost of each victim-hospital assignment (calculateCost Function) that falls out of the proposed final allocation), and the first three bars in the Egalitarian section show the final value calculated with the Egalitarian social welfare function (i.e. the worst off value selected from what falls out of the proposed final allocation) (when running Utilitarian methods to perform the local search). Moreover, the last three bars in the initial section show the initial value calculated with the Egalitarian social welfare function, the last three bars in the Utilitarian section show the final value calculated with the Utilitarian social welfare function (when running Egalitarian methods to perform the local search), and the last three bars in the Egalitarian section show the final value calculated with the Egalitarian social welfare function. The initial cost is only positive for even distribution (Utilitarian) and random distribution (Utilitarian). After running the local search in the Utilitarian approach, all the Utilitarian results even out by only differing by 4 with random distribution (Utilitarian) having the highest total cost. Looking at the Egalitarian results, the single distribution (Egalitarian) has the most but it is important to note that these values are from running the Egalitarian and they are clearly less than the Utilitarian results. The results from the Egalitarian approach seem to be very even with single distribution (Egalitarian) as the best score. However, the difference between first and last is less than 3. Based on total cost of both approaches, it seems even distribution (Utilitarian) is a good choice.

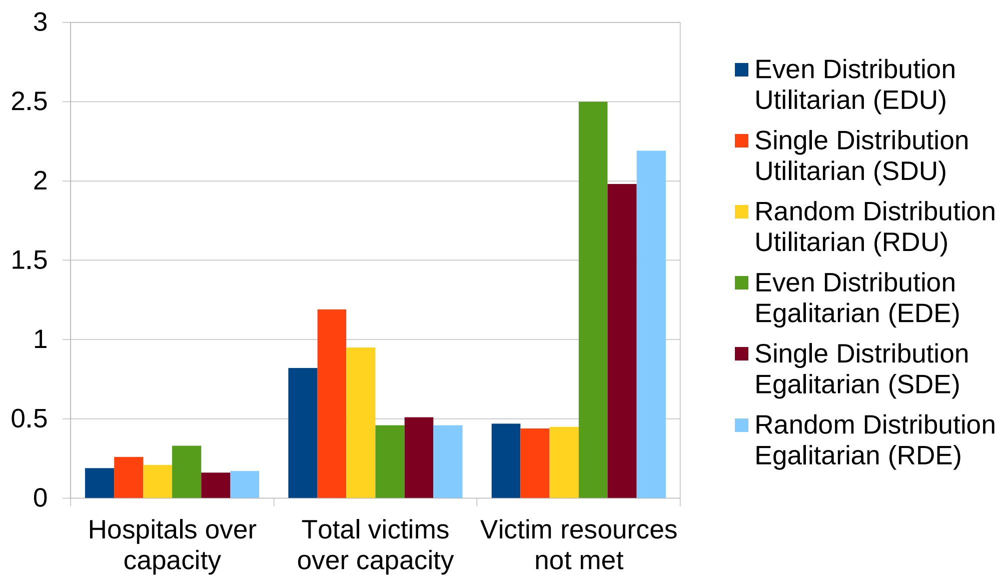

Figure A.2 – Constraints graph

Figure A.2 shows a graph based on the soft constraints. It maps the extent to which hospitals are over capacity, the total victims over capacity, and victim resources are not met. Here, a high number reflects a lower overall value to the victim, which is undesirable. The positive note from this graph is that on average less than 2 victims are over capacity and less than 3 victims do not have their resources met. This ensures that the hospitals are efficiently treating their patients. However, Table A.1 displays a different note with a table that represents the worst run. This table shows that the total capacity can be as high as 13, for the Utilitarian approaches, and the number of victims that did not have their resources met also is 13, for single distribution (Egalitarian). This result could just be a bad run where each hospital had low resources available and low capacity. It is a reasonable situation since even distribution (Egalitarian) had all three hospitals over the capacity. Table A.1 also includes the maximum number of iterations performed during the search. Both even distribution approaches have the least amount of maximum iterations.

Figure A.3 shows a graph based on the number of iterations. Iteration can either be swapping a hospital between victims or a victim being assigned another hospital. Note that here, a lower number of iterations provides value to the victims, as long as the allocation selected is one where the result's total cost is desirable. Single distribution approaches have the most iteration because of the initial state having all the victims in the same hospital. This results in many new assignments of victims to another hospital. In contrast the other distributions tend to swap hospitals between patients more often on average. Note that here, a lower number provides value to the victims, as long as the allocation selected is one where the result's total cost is desirable. Based on the experimental results, it seems that random distribution (Utilitarian) is the best approach for utilitarian. This is due to its high total cost, few constraints broken and least amount of iterations. Single distribution (Egalitarian) approach is the best for the egalitarian. Even though the maximum number of iterations is high and constraints statistics are average, the total costs are the best for both kinds of social welfare. It is important to note that random distribution (Utilitarian) had a very good result for the egalitarian and therefore should be considered best approach overall. Some results not displayed in the graphs were obtained using single distribution (Utilitarian) in three other scenarios. They included 20 patients with 2 hospitals, 40 patients with 4 hospitals, and 50 patients with 5 hospitals as well as 30 patients with 3 hospitals. The scenario with 20 patients had the best total cost per iterations but more experimentation is needed for these scenarios. We note as well that, through all our experiments, the time taken to complete the runs of the scenarios were all quite low (e.g. on the order of 172 milliseconds for 100 scenarios tested with EDU). We therefore are encouraged that this algorithm may be of value towards real deployment (where various values required such as skill of the EMS driver would already be pre-stored).

Table A.1 – Maximum results table

	Max Iterations	Max Hospitals Over Capacity	Max Total Capacity	Max Victim Resources Not Met
Even distribution (Utilitarian)	36	2	13	11
Single distribution (Utilitarian)	79	1	13	11
Average random (Utilitarian)	37	1	13	11
Even distribution (Egalitarian)	12	3	6	11
Single distribution (Egalitarian)	65	1	6	13
Average random (Egalitarian)	14	2	6	12

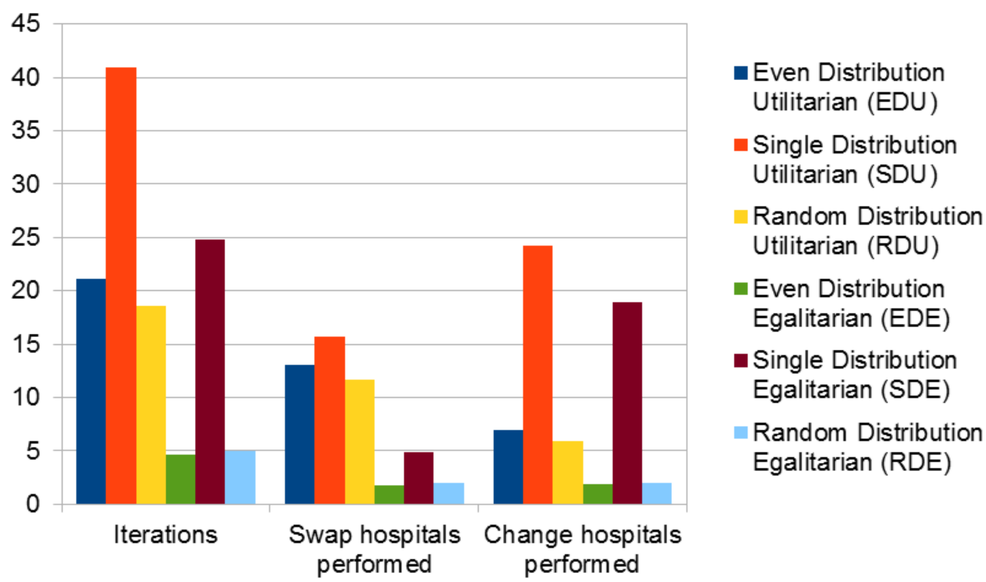

Figure A.3 – Iterations graph

A.4 Summary

We have been able to show how to take examples of mass casualty incidents and convert them to constraint satisfaction problems. The implementation of this problem into the Java programming language allowed this problem to be solved using a local search. The results of a local search will vary based on the implementation, which lead to experimenting with different approaches. Adding a global view provided an interesting consideration when assigning a patient to a hospital. Overall, the random distribution (Utilitarian) approach seemed to have the best results.

Appendix B
A Look at the Cost Calculations

In this appendix, we discuss in further detail the range of possible values for an allocation (i.e. the total costs that could be incurred). We use as an example the Utilitarian social welfare function. We do so both to clarify how ranges of parameter values were chosen when we modeled the Ambulance and Hospital phases (Section 4.2) and to help the reader understand some of the values displayed on the axis of our graphs, in the Implementation Output Chapter (Chapter 5).

The penalty values will be set as follows: LOWRC (2), HIGHRC (10), noAmbulanceDriveCost (18).

The allocation value (i.e. the total cost of the allocation, computed per Algorithm 4.2) is derived on the basis of adding the costs incurred from each SingleSolution allocation (i.e. what each victim receives as its resource (ambulance, hospital pair) and its cost (per the CostAmbulance Function)). We will sketch this sum computed over 30 victims, 20 ambulances, and 3 hospitals (the number used in our implementation (see chapter 5)) for the Initial phase. The total cost also adds in some possible penalties for ambulance (adding up costs incurred for each ambulance) and hospital (adding up costs incurred for each hospital). The range of possible values for the total cost turns out to be [364,970]. This can be determined as follows.

In order to see the possible values, it is useful to view the parameters listed in Tables 4.1, 4.2, and 4.3 and their indicated upper and lower bounds for values. Each SingleSolution allocation (determined on the basis of the CostAmbulance Function) is at **least** as follows: cost of drive turns out to be at least 2, cost of victim wait is at least 1, cost of resource is at least 2 so that the cost for this single solution overall is at least 5. When a resource is null, the SingleSolution allocation is as follows: cost of drive turns out to be at least 14, cost of victim wait is at least 1, cost of resource is at least 10 so that the cost for this single solution overall is at least 25.

Cost of wait has upper and lower values as indicated in the tables. Cost of drive is determined by the CostDrive Function which adds together (minimum values per the tables indicated in parentheses)

1 + hdist (1) + (maxVictim - vicSev) * (maxType - ambType) (where maxType - ambType = (2-2) so this expression is at min 0) + (maxType * (vicSev -1)) (where vicSev – 1 = (1-1) so this expression is at min 0) is at least 2 for cost of drive (1 + 1). Cost of resource is determined by the CostRes

Function, which returns (minimum values per the tables indicated in parentheses) hRes – vRes (where hRes = vRes so this expression is at min 0) + LOWRC is at least 2.

Each SingleSolution allocation (determined on the basis of the CostAmbulance Function) is at **most** as follows: cost of drive turns out to be at most 17, cost of victim wait is at most 5, cost of resource is at most 10 so that the cost for this single solution overall is at most 32. This value occurs when a resource is null. When a resource is not null, the SingleSolution allocation is as follows: cost of drive turns out to be at most 13, cost of victim wait is at most 5, cost of resource is at most 6 so that the cost for this single solution overall is at most 24.

Cost of wait has upper and lower values as indicated in Tables 4.1, 4.2, and 4.3. Cost of drive is determined by the CostDrive Function which adds together (maximum values per Tables 4.1, 4.2, and 4.3 indicated in parentheses)

noAmbulanceDriveCost – vicSev (18 – 1) is at most 17 for cost of drive. Cost of resource is determined by the CostRes Function, which returns HIGHRC is at most10.

For the 30 victims, each assignment value can range from 5 to 32 meaning that when using the Utilitarian social welfare function (which adds all the individual allocation values) the minimum is 150 and the maximum is 960. We can tighten the upper limit since we know that 20 of the victims receive a resource (since there are 20 ambulances (which are not shareable)). With these victims receiving the highest cost resources, the maximum becomes 800. This also allows for a tighter lower limit taking into account that 10 victims will not receive a resource. The minimum will become 340.

Each ambulance is allocated to a victim. If the ambulance is a null ambulance then it means that there is a victim did not receive an ambulance. A cost of 2 is incurred for each ambulance that is a null ambulance. Since there are 20 ambulances for the 30 victims, the cost would be 20.

Each hospital incurs a cost based its capacity, which can either be 8, 12, or 15 (as indicated by Table 4.3). A hospital incurs a cost of 10 for each victim over capacity and a cost of 1 for each victim under capacity. Since the 20 ambulances are driving to 3 hospitals, the minimum cost is 4 and maximum is 150.

Note that Figure VA-1 (a), which plots allocation values that arose from simulating an ambulance environment ended up with various options each roughly incurring a total cost between 520 and 590; this is within the maximal possible range demonstrated above, 340 and 800.

Appendix C
Decision Support for Ambulance and Hospital Stages

This section provides an example of decision support by showing the available resources being assigned to patients and the resulting allocation value. There are two solutions used to represent a possible system and user allocation. Each stage provides an example of patients and resources (including their attribute values (where the value for each principle of allocation attribute represents that patient's rank in the set of patients)), as well as the allocation and the allocation value for the system and user recommended allocations. The percentage change is calculated for these two solutions.

Ambulance Stage Decision Support

The values for the attributes were generated from Test 90 of the Ambulance stage with 30 victims, 20 ambulances, 3 hospitals, and at time step 0. The allocation values were generated using Utilitarian social welfare and no principle policy. The system allocation used local search with 100 random restarts with Utilitarian social welfare and no principle policy. The user recommended allocation used the principle of allocation approach using prognosis as the principle.

Victims

ID	Values
1	severity=2, resources=3, initialTime=0, availability=0, lottery=19, fcfs=18, sickestFirst=6, youngestFirst=21, livesSaved=21, prognosis=21, instrumental=11, reciprocity=18, age=56, survivalPerctentage=39, expectedYears=19
2	severity=3, resources=1, initialTime=0, availability=0, lottery=3, fcfs=25, sickestFirst=21, youngestFirst=14, livesSaved=12, prognosis=12, instrumental=23, reciprocity=25, age=41, survivalPerctentage=68, expectedYears=34
3	severity=3, resources=1, initialTime=0, availability=0, lottery=12, fcfs=1, sickestFirst=20, youngestFirst=22, livesSaved=14, prognosis=14, instrumental=3, reciprocity=1, age=58, survivalPerctentage=63, expectedYears=31
4	severity=4, resources=2, initialTime=0, availability=0, lottery=11, fcfs=11, sickestFirst=25, youngestFirst=25, livesSaved=1, prognosis=1, instrumental=12, reciprocity=11, age=65, survivalPerctentage=91, expectedYears=45
5	severity=1, resources=1, initialTime=0, availability=0, lottery=24, fcfs=23, sickestFirst=3,

	youngestFirst=23, livesSaved=28, prognosis=28, instrumental=17, reciprocity=23, age=61, survivalPerctentage=16, expectedYears=8
6	severity=4, resources=2, initialTime=0, availability=0, lottery=0, fcfs=14, sickestFirst=26, youngestFirst=19, livesSaved=7, prognosis=7, instrumental=25, reciprocity=14, age=49, survivalPerctentage=73, expectedYears=36
7	severity=1, resources=1, initialTime=0, availability=0, lottery=10, fcfs=28, sickestFirst=4, youngestFirst=26, livesSaved=26, prognosis=26, instrumental=16, reciprocity=28, age=66, survivalPerctentage=24, expectedYears=12
8	severity=3, resources=2, initialTime=0, availability=0, lottery=14, fcfs=29, sickestFirst=19, youngestFirst=3, livesSaved=5, prognosis=5, instrumental=19, reciprocity=29, age=24, survivalPerctentage=77, expectedYears=38
9	severity=4, resources=1, initialTime=0, availability=0, lottery=25, fcfs=4, sickestFirst=27, youngestFirst=17, livesSaved=9, prognosis=9, instrumental=26, reciprocity=4, age=46, survivalPerctentage=72, expectedYears=36
10	severity=1, resources=3, initialTime=0, availability=0, lottery=8, fcfs=6, sickestFirst=1, youngestFirst=7, livesSaved=24, prognosis=24, instrumental=24, reciprocity=6, age=28, survivalPerctentage=34, expectedYears=17
11	severity=4, resources=3, initialTime=0, availability=0, lottery=21, fcfs=21, sickestFirst=23, youngestFirst=9, livesSaved=2, prognosis=2, instrumental=5, reciprocity=21, age=29, survivalPerctentage=87, expectedYears=43
12	severity=3, resources=3, initialTime=0, availability=0, lottery=6, fcfs=22, sickestFirst=15, youngestFirst=13, livesSaved=13, prognosis=13, instrumental=0, reciprocity=22, age=36, survivalPerctentage=65, expectedYears=32
13	severity=3, resources=2, initialTime=0, availability=0, lottery=20, fcfs=5, sickestFirst=17, youngestFirst=27, livesSaved=16, prognosis=16, instrumental=13, reciprocity=5, age=66, survivalPerctentage=56, expectedYears=28
14	severity=2, resources=2, initialTime=0, availability=0, lottery=26, fcfs=10, sickestFirst=8, youngestFirst=29, livesSaved=20, prognosis=20, instrumental=20, reciprocity=10, age=69, survivalPerctentage=41, expectedYears=20
15	severity=1, resources=2, initialTime=0, availability=0, lottery=13, fcfs=2, sickestFirst=2, youngestFirst=24, livesSaved=29, prognosis=29, instrumental=2, reciprocity=2, age=63,

	survivalPerctentage=12, expectedYears=6
16	severity=3, resources=3, initialTime=0, availability=0, lottery=2, fcfs=26, sickestFirst=16, youngestFirst=1, livesSaved=15, prognosis=15, instrumental=4, reciprocity=26, age=23, survivalPerctentage=59, expectedYears=29
17	severity=4, resources=2, initialTime=0, availability=0, lottery=29, fcfs=9, sickestFirst=24, youngestFirst=0, livesSaved=3, prognosis=3, instrumental=1, reciprocity=9, age=21, survivalPerctentage=78, expectedYears=39
18	severity=4, resources=1, initialTime=0, availability=0, lottery=18, fcfs=16, sickestFirst=29, youngestFirst=8, livesSaved=0, prognosis=0, instrumental=27, reciprocity=16, age=28, survivalPerctentage=97, expectedYears=48
19	severity=2, resources=2, initialTime=0, availability=0, lottery=9, fcfs=19, sickestFirst=9, youngestFirst=28, livesSaved=18, prognosis=18, instrumental=6, reciprocity=19, age=67, survivalPerctentage=51, expectedYears=25
20	severity=3, resources=1, initialTime=0, availability=0, lottery=1, fcfs=27, sickestFirst=22, youngestFirst=2, livesSaved=6, prognosis=6, instrumental=8, reciprocity=27, age=23, survivalPerctentage=75, expectedYears=37
21	severity=2, resources=1, initialTime=0, availability=0, lottery=17, fcfs=15, sickestFirst=11, youngestFirst=5, livesSaved=19, prognosis=19, instrumental=9, reciprocity=15, age=27, survivalPerctentage=42, expectedYears=21
22	severity=3, resources=3, initialTime=0, availability=0, lottery=22, fcfs=8, sickestFirst=13, youngestFirst=16, livesSaved=11, prognosis=11, instrumental=15, reciprocity=8, age=44, survivalPerctentage=69, expectedYears=34
23	severity=2, resources=2, initialTime=0, availability=0, lottery=7, fcfs=20, sickestFirst=10, youngestFirst=4, livesSaved=25, prognosis=25, instrumental=10, reciprocity=20, age=25, survivalPerctentage=33, expectedYears=16
24	severity=2, resources=1, initialTime=0, availability=0, lottery=15, fcfs=24, sickestFirst=12, youngestFirst=11, livesSaved=23, prognosis=23, instrumental=7, reciprocity=24, age=36, survivalPerctentage=36, expectedYears=18
25	severity=2, resources=3, initialTime=0, availability=0, lottery=4, fcfs=0, sickestFirst=5, youngestFirst=18, livesSaved=22, prognosis=22, instrumental=29, reciprocity=0, age=48, survivalPerctentage=38, expectedYears=19

26	severity=3, resources=3, initialTime=0, availability=0, lottery=27, fcfs=12, sickestFirst=14, youngestFirst=12, livesSaved=10, prognosis=10, instrumental=18, reciprocity=12, age=36, survivalPerctentage=71, expectedYears=35
27	severity=2, resources=2, initialTime=0, availability=0, lottery=16, fcfs=7, sickestFirst=7, youngestFirst=15, livesSaved=17, prognosis=17, instrumental=22, reciprocity=7, age=44, survivalPerctentage=52, expectedYears=26
28	severity=3, resources=2, initialTime=0, availability=0, lottery=28, fcfs=17, sickestFirst=18, youngestFirst=10, livesSaved=8, prognosis=8, instrumental=28, reciprocity=17, age=34, survivalPerctentage=72, expectedYears=36
29	severity=4, resources=1, initialTime=0, availability=0, lottery=5, fcfs=13, sickestFirst=28, youngestFirst=20, livesSaved=4, prognosis=4, instrumental=21, reciprocity=13, age=54, survivalPerctentage=78, expectedYears=39
30	severity=1, resources=3, initialTime=0, availability=0, lottery=23, fcfs=3, sickestFirst=0, youngestFirst=6, livesSaved=27, prognosis=27, instrumental=14, reciprocity=3, age=28, survivalPerctentage=18, expectedYears=9

Ambulance

ID	**Values**
34	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
35	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
36	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
37	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
38	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
39	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
40	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
41	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
42	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
43	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
44	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
45	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
46	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
47	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false

48	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
49	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
50	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
51	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
52	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
53	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false

Hospital

ID	**Values**
31	resources=1, currentPatients=0, capacity=8, distance=2, longitude=3, latitude=-4
32	resources=2, currentPatients=0, capacity=12, distance=1, longitude=-3, latitude=0
33	resources=2, currentPatients=0, capacity=12, distance=2, longitude=1, latitude=4

System allocation – allocation value = 525

Patient ID	**Ambulance ID**	**Hospital ID**
1	36	33
2	44	31
3	45	31
4	Null Resource	Null Resource
5	34	31
6	Null Resource	Null Resource
7	37	32
8	50	32
9	Null Resource	Null Resource
10	53	32
11	Null Resource	Null Resource
12	Null Resource	Null Resource
13	27	32
14	35	32
15	38	32
16	Null Resource	Null Resource
17	Null Resource	Null Resource

18	Null Resource	Null Resource
19	39	32
20	51	31
21	52	31
22	Null Resource	Null Resource
23	40	32
24	43	31
25	47	32
26	Null Resource	Null Resource
27	48	32
28	41	32
29	42	31
30	49	32

User allocation – allocation value = 588

Patient ID	**Ambulance ID**	**Hospital ID**
1	Null Resource	Null Resource
2	53	31
3	45	31
4	35	32
5	Null Resource	Null Resource
6	41	32
7	Null Resource	Null Resource
8	39	32
9	42	31
10	Null Resource	Null Resource
11	36	32
12	44	32
13	48	32
14	Null Resource	Null Resource
15	Null Resource	Null Resource

16	46	32
17	37	32
18	34	31
19	51	33
20	40	31
21	52	31
22	49	32
23	Null Resource	Null Resource
24	Null Resource	Null Resource
25	Null Resource	Null Resource
26	47	32
27	50	32
28	43	32
29	38	31
30	Null Resource	Null Resource

Percentage Change between system allocation and user allocation = ((525-588)/525)*100 = -12.00%

Hospital Stage Decision Support

The values for the attributes were generated from Test 0 of the Hospital stage with 10 patients, 6 resources, and at time step 0. The allocation values were generated using Utilitarian social welfare and no principle policy. The system allocation used local search with 100 random restarts with Utilitarian social welfare and no principle policy. The user recommended allocation used the principle of allocation approach using prognosis as the principle.

Patients

ID	**Values**
1	severity=2, resources=1, initialTime=0, availability=0, lottery=0, fcfs=7, sickestFirst=4, youngestFirst=2, livesSaved=6, prognosis=6, instrumental=6, reciprocity=7, age=43, survivalPerctentage=45, expectedYears=22
2	severity=2, resources=2, initialTime=0, availability=0, lottery=1, fcfs=8, sickestFirst=3, youngestFirst=9, livesSaved=5, prognosis=5, instrumental=1, reciprocity=8, age=69, survivalPerctentage=48, expectedYears=24

3	severity=2, resources=3, initialTime=0, availability=0, lottery=8, fcfs=0, sickestFirst=2, youngestFirst=7, livesSaved=7, prognosis=7, instrumental=3, reciprocity=0, age=61, survivalPerctentage=41, expectedYears=20
4	severity=3, resources=2, initialTime=0, availability=0, lottery=3, fcfs=3, sickestFirst=5, youngestFirst=1, livesSaved=4, prognosis=4, instrumental=4, reciprocity=3, age=28, survivalPerctentage=62, expectedYears=31
5	severity=4, resources=3, initialTime=0, availability=0, lottery=5, fcfs=6, sickestFirst=8, youngestFirst=8, livesSaved=2, prognosis=2, instrumental=0, reciprocity=6, age=65, survivalPerctentage=81, expectedYears=40
6	severity=4, resources=1, initialTime=0, availability=0, lottery=7, fcfs=1, sickestFirst=9, youngestFirst=4, livesSaved=1, prognosis=1, instrumental=8, reciprocity=1, age=50, survivalPerctentage=88, expectedYears=44
7	severity=4, resources=3, initialTime=0, availability=0, lottery=9, fcfs=2, sickestFirst=7, youngestFirst=5, livesSaved=0, prognosis=0, instrumental=7, reciprocity=2, age=56, survivalPerctentage=89, expectedYears=44
8	severity=1, resources=3, initialTime=0, availability=0, lottery=4, fcfs=9, sickestFirst=0, youngestFirst=0, livesSaved=8, prognosis=8, instrumental=9, reciprocity=9, age=23, survivalPerctentage=29, expectedYears=14
9	severity=4, resources=4, initialTime=0, availability=0, lottery=6, fcfs=5, sickestFirst=6, youngestFirst=3, livesSaved=3, prognosis=3, instrumental=5, reciprocity=5, age=43, survivalPerctentage=74, expectedYears=37
10	severity=1, resources=1, initialTime=0, availability=0, lottery=2, fcfs=4, sickestFirst=1, youngestFirst=6, livesSaved=9, prognosis=9, instrumental=2, reciprocity=4, age=58, survivalPerctentage=16, expectedYears=8

Resources

ID	**Values**
11	type=2, initialTime=0, availability=0
12	type=5, initialTime=0, availability=0
13	type=1, initialTime=0, availability=0
14	type=2, initialTime=0, availability=0
15	type=3, initialTime=0, availability=0

16	type=3, initialTime=0, availability=0

System allocation – allocation value = 289

Patient ID	**Resource ID**
1	11
2	14
3	15
4	13
5	Null Resource
6	Null Resource
7	Null Resource
8	16
9	Null Resource
10	12

User allocation – allocation value = 336

Patient ID	**Resource ID**
1	Null Resource
2	13
3	Null Resource
4	14
5	16
6	15
7	12
8	Null Resource
9	11
10	Null Resource

Percentage Change between system allocation and user allocation = ((289-336)/289)*100 = -16.26%

Appendix D

Trace of Local Search for Ambulance and Hospital Stages

This section provides a trace during the local search. The local search uses 100 random restarts and the trace represents an allocation that had the best allocation value at the end of the local search. Each stage provides an example of patients and resources (including their attribute values (where the value for each principle of allocation attribute represents that patient's rank in the set of patients)). The initial assignment of resources and patients is shown along with the allocation value. Each new assignment represents the better neighbor selected during the local search. The final assignment represents the solution as none of the neighbors are better than the current solution.

<u>Trace Ambulance Stage</u>

The Ambulance stage example has 15 victims, 10 ambulances, 3 hospitals, and at time step 0. The allocation values were generated using Utilitarian social welfare and no principle policy.

Victims

ID	**Values**
1	severity=3, resources=1, initialTime=0, availability=0, lottery=5, fcfs=0, sickestFirst=9, youngestFirst=2, livesSaved=4, prognosis=4, instrumental=0, reciprocity=0, age=33, survivalPerctentage=65, expectedYears=32
2	severity=4, resources=2, initialTime=0, availability=0, lottery=14, fcfs=7, sickestFirst=13, youngestFirst=5, livesSaved=0, prognosis=0, instrumental=10, reciprocity=7, age=50, survivalPerctentage=88, expectedYears=44
3	severity=1, resources=2, initialTime=0, availability=0, lottery=1, fcfs=9, sickestFirst=0, youngestFirst=10, livesSaved=13, prognosis=13, instrumental=11, reciprocity=9, age=63, survivalPerctentage=20, expectedYears=10
4	severity=3, resources=1, initialTime=0, availability=0, lottery=12, fcfs=1, sickestFirst=10, youngestFirst=8, livesSaved=8, prognosis=8, instrumental=8, reciprocity=1, age=60, survivalPerctentage=54, expectedYears=27
5	severity=3, resources=3, initialTime=0, availability=0, lottery=11, fcfs=12, sickestFirst=6, youngestFirst=12, livesSaved=7, prognosis=7, instrumental=3, reciprocity=12, age=66, survivalPerctentage=57, expectedYears=28

6	severity=2, resources=2, initialTime=0, availability=0, lottery=7, fcfs=5, sickestFirst=3, youngestFirst=0, livesSaved=12, prognosis=12, instrumental=6, reciprocity=5, age=26, survivalPerctentage=40, expectedYears=20
7	severity=3, resources=3, initialTime=0, availability=0, lottery=10, fcfs=14, sickestFirst=7, youngestFirst=13, livesSaved=5, prognosis=5, instrumental=4, reciprocity=14, age=67, survivalPerctentage=64, expectedYears=32
8	severity=1, resources=2, initialTime=0, availability=0, lottery=4, fcfs=13, sickestFirst=1, youngestFirst=7, livesSaved=14, prognosis=14, instrumental=14, reciprocity=13, age=58, survivalPerctentage=18, expectedYears=9
9	severity=3, resources=2, initialTime=0, availability=0, lottery=0, fcfs=4, sickestFirst=8, youngestFirst=6, livesSaved=3, prognosis=3, instrumental=5, reciprocity=4, age=51, survivalPerctentage=68, expectedYears=34
10	severity=4, resources=2, initialTime=0, availability=0, lottery=6, fcfs=2, sickestFirst=12, youngestFirst=3, livesSaved=2, prognosis=2, instrumental=12, reciprocity=2, age=34, survivalPerctentage=79, expectedYears=39
11	severity=2, resources=2, initialTime=0, availability=0, lottery=2, fcfs=3, sickestFirst=2, youngestFirst=4, livesSaved=10, prognosis=10, instrumental=13, reciprocity=3, age=43, survivalPerctentage=45, expectedYears=22
12	severity=2, resources=2, initialTime=0, availability=0, lottery=9, fcfs=6, sickestFirst=4, youngestFirst=14, livesSaved=9, prognosis=9, instrumental=1, reciprocity=6, age=69, survivalPerctentage=48, expectedYears=24
13	severity=2, resources=2, initialTime=0, availability=0, lottery=13, fcfs=8, sickestFirst=5, youngestFirst=9, livesSaved=11, prognosis=11, instrumental=2, reciprocity=8, age=61, survivalPerctentage=41, expectedYears=20
14	severity=3, resources=1, initialTime=0, availability=0, lottery=3, fcfs=11, sickestFirst=11, youngestFirst=1, livesSaved=6, prognosis=6, instrumental=7, reciprocity=11, age=28, survivalPerctentage=62, expectedYears=31
15	severity=4, resources=1, initialTime=0, availability=0, lottery=8, fcfs=10, sickestFirst=14, youngestFirst=11, livesSaved=1, prognosis=1, instrumental=9, reciprocity=10, age=65, survivalPerctentage=81, expectedYears=40

Ambulance

ID	Values
19	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
20	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
21	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
22	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
23	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
24	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
25	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
26	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
27	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
28	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false

Hospital

ID	Values
16	resources=3, currentPatients=0, capacity=15, distance=2, longitude=-2, latitude=4
17	resources=2, currentPatients=0, capacity=12, distance=3, longitude=-4, latitude=4
18	resources=1, currentPatients=0, capacity=15, distance=1, longitude=3, latitude=0

Allocation Value: 312.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	24	16
10	19	17
3	Null Resource	Null Resource
9	28	18
11	23	17
5	27	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 311.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	24	17
10	19	17
3	Null Resource	Null Resource
9	28	18
11	23	17
5	27	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 310.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	19	17
3	Null Resource	Null Resource
9	28	18
11	24	17
5	27	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 305.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	19	16
9	28	18
11	24	17
5	27	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 304.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	19	17
9	28	18
11	24	17
5	27	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 303.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	19	17
9	28	16
11	24	17
5	27	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 301.0		
Victim	Ambulance	Hospital

Allocation Value: 300.0		
Victim	Ambulance	Hospital

7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	27	17
9	28	16
11	24	17
5	19	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	27	17
9	28	17
11	24	17
5	19	17
4	Null Resource	Null Resource
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 299.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	27	17
9	Null Resource	Null Resource
11	24	17
5	19	17
4	28	18
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 296.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource
3	27	17
9	19	17
11	24	17
5	Null Resource	Null Resource
4	28	18
1	20	18
6	21	17
12	Null Resource	Null Resource
15	Null Resource	Null Resource
13	22	16
2	26	16
14	25	18

Allocation Value: 295.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource

Allocation Value: 292.0		
Victim	Ambulance	Hospital
7	Null Resource	Null Resource
8	23	17
10	Null Resource	Null Resource

3	27	17		3	27	17
9	19	17		9	19	17
11	24	17		11	24	17
5	Null Resource	Null Resource		5	26	16
4	28	18		4	28	18
1	20	18		1	20	18
6	21	17		6	21	17
12	Null Resource	Null Resource		12	Null Resource	Null Resource
15	Null Resource	Null Resource		15	Null Resource	Null Resource
13	22	17		13	22	17
2	26	16		2	Null Resource	Null Resource
14	25	18		14	25	18

Trace Hospital Stage

The values for the attributes were generated from Test 0 of the Hospital stage with 10 patients, 6 resources, and at time step 0. The allocation values were generated using Utilitarian social welfare and no principle policy.

Patients

ID	**Values**
1	severity=2, resources=1, initialTime=0, availability=0, lottery=0, fcfs=7, sickestFirst=4, youngestFirst=2, livesSaved=6, prognosis=6, instrumental=6, reciprocity=7, age=43, survivalPerctentage=45, expectedYears=22
2	severity=2, resources=2, initialTime=0, availability=0, lottery=1, fcfs=8, sickestFirst=3, youngestFirst=9, livesSaved=5, prognosis=5, instrumental=1, reciprocity=8, age=69, survivalPerctentage=48, expectedYears=24
3	severity=2, resources=3, initialTime=0, availability=0, lottery=8, fcfs=0, sickestFirst=2, youngestFirst=7, livesSaved=7, prognosis=7, instrumental=3, reciprocity=0, age=61, survivalPerctentage=41, expectedYears=20
4	severity=3, resources=2, initialTime=0, availability=0, lottery=3, fcfs=3, sickestFirst=5, youngestFirst=1, livesSaved=4, prognosis=4, instrumental=4, reciprocity=3, age=28, survivalPerctentage=62, expectedYears=31
5	severity=4, resources=3, initialTime=0, availability=0, lottery=5, fcfs=6, sickestFirst=8, youngestFirst=8, livesSaved=2, prognosis=2, instrumental=0, reciprocity=6, age=65, survivalPerctentage=81, expectedYears=40

6	severity=4, resources=1, initialTime=0, availability=0, lottery=7, fcfs=1, sickestFirst=9, youngestFirst=4, livesSaved=1, prognosis=1, instrumental=8, reciprocity=1, age=50, survivalPerctentage=88, expectedYears=44
7	severity=4, resources=3, initialTime=0, availability=0, lottery=9, fcfs=2, sickestFirst=7, youngestFirst=5, livesSaved=0, prognosis=0, instrumental=7, reciprocity=2, age=56, survivalPerctentage=89, expectedYears=44
8	severity=1, resources=3, initialTime=0, availability=0, lottery=4, fcfs=9, sickestFirst=0, youngestFirst=0, livesSaved=8, prognosis=8, instrumental=9, reciprocity=9, age=23, survivalPerctentage=29, expectedYears=14
9	severity=4, resources=4, initialTime=0, availability=0, lottery=6, fcfs=5, sickestFirst=6, youngestFirst=3, livesSaved=3, prognosis=3, instrumental=5, reciprocity=5, age=43, survivalPerctentage=74, expectedYears=37
10	severity=1, resources=1, initialTime=0, availability=0, lottery=2, fcfs=4, sickestFirst=1, youngestFirst=6, livesSaved=9, prognosis=9, instrumental=2, reciprocity=4, age=58, survivalPerctentage=16, expectedYears=8

Resources

ID	**Values**
11	type=2, initialTime=0, availability=0
12	type=5, initialTime=0, availability=0
13	type=1, initialTime=0, availability=0
14	type=2, initialTime=0, availability=0
15	type=3, initialTime=0, availability=0
16	type=3, initialTime=0, availability=0

Allocation Value: 308	
Patient	Resource
1	12
2	16
3	15
4	Null Resource
5	11

Allocation Value: 306	
Patient	Resource
1	12
2	16
3	13
4	Null Resource
5	11

Allocation Value: 305	
Patient	Resource
1	15
2	16
3	13
4	Null Resource
5	11

Allocation Value: 304	
Patient	Resource
1	15
2	16
3	13
4	Null Resource
5	11

6	Null Resource
7	Null Resource
8	13
9	14
10	Null Resource

6	Null Resource
7	Null Resource
8	15
9	14
10	Null Resource

6	Null Resource
7	Null Resource
8	12
9	14
10	Null Resource

6	Null Resource
7	Null Resource
8	Null Resource
9	14
10	12

Allocation Value: 302	
Patient	Resource
1	15
2	16
3	Null Resource
4	Null Resource
5	11
6	Null Resource
7	Null Resource
8	13
9	14
10	12

Allocation Value: 298	
Patient	Resource
1	15
2	16
3	Null Resource
4	Null Resource
5	11
6	Null Resource
7	Null Resource
8	14
9	13
10	12

Allocation Value: 297	
Patient	Resource
1	15
2	14
3	Null Resource
4	Null Resource
5	11
6	Null Resource
7	Null Resource
8	16
9	13
10	12

Allocation Value: 294	
Patient	Resource
1	15
2	14
3	13
4	Null Resource
5	11
6	Null Resource
7	Null Resource
8	16
9	Null Resource
10	12

Allocation Value: 292	
Patient	Resource
1	15
2	14
3	11
4	Null Resource
5	13
6	Null Resource
7	Null Resource
8	16

Allocation Value: 291	
Patient	Resource
1	11
2	14
3	15
4	Null Resource
5	13
6	Null Resource
7	Null Resource
8	16

Allocation Value: 290	
Patient	Resource
1	11
2	14
3	15
4	Null Resource
5	Null Resource
6	Null Resource
7	Null Resource
8	16

Allocation Value: 289	
Patient	Resource
1	11
2	14
3	15
4	13
5	Null Resource
6	Null Resource
7	Null Resource
8	16

9	Null Resource
10	12

9	Null Resource
10	12

9	13
10	12

9	Null Resource
10	12

Appendix E

Example of Implementing Different Random Restart Values for Local Search

This section provides an example of the different allocation using different stopping criteria during the local search. There are two solutions used to represent 10 and 1 random restart. The values for the attributes (where the value for each principle of allocation attribute represents that patient's rank in the set of patients) were generated from Test 90 of the Ambulance stage with 30 victims, 20 ambulances, 3 hospitals, and at time step 0. The allocation values were generated using Utilitarian social welfare and no principle policy.

Victims

ID	**Values**
1	severity=2, resources=3, initialTime=0, availability=0, lottery=19, fcfs=18, sickestFirst=6, youngestFirst=21, livesSaved=21, prognosis=21, instrumental=11, reciprocity=18, age=56, survivalPerctentage=39, expectedYears=19
2	severity=3, resources=1, initialTime=0, availability=0, lottery=3, fcfs=25, sickestFirst=21, youngestFirst=14, livesSaved=12, prognosis=12, instrumental=23, reciprocity=25, age=41, survivalPerctentage=68, expectedYears=34
3	severity=3, resources=1, initialTime=0, availability=0, lottery=12, fcfs=1, sickestFirst=20, youngestFirst=22, livesSaved=14, prognosis=14, instrumental=3, reciprocity=1, age=58, survivalPerctentage=63, expectedYears=31
4	severity=4, resources=2, initialTime=0, availability=0, lottery=11, fcfs=11, sickestFirst=25, youngestFirst=25, livesSaved=1, prognosis=1, instrumental=12, reciprocity=11, age=65, survivalPerctentage=91, expectedYears=45
5	severity=1, resources=1, initialTime=0, availability=0, lottery=24, fcfs=23, sickestFirst=3, youngestFirst=23, livesSaved=28, prognosis=28, instrumental=17, reciprocity=23, age=61, survivalPerctentage=16, expectedYears=8
6	severity=4, resources=2, initialTime=0, availability=0, lottery=0, fcfs=14, sickestFirst=26, youngestFirst=19, livesSaved=7, prognosis=7, instrumental=25, reciprocity=14, age=49, survivalPerctentage=73, expectedYears=36

7	severity=1, resources=1, initialTime=0, availability=0, lottery=10, fcfs=28, sickestFirst=4, youngestFirst=26, livesSaved=26, prognosis=26, instrumental=16, reciprocity=28, age=66, survivalPerctentage=24, expectedYears=12
8	severity=3, resources=2, initialTime=0, availability=0, lottery=14, fcfs=29, sickestFirst=19, youngestFirst=3, livesSaved=5, prognosis=5, instrumental=19, reciprocity=29, age=24, survivalPerctentage=77, expectedYears=38
9	severity=4, resources=1, initialTime=0, availability=0, lottery=25, fcfs=4, sickestFirst=27, youngestFirst=17, livesSaved=9, prognosis=9, instrumental=26, reciprocity=4, age=46, survivalPerctentage=72, expectedYears=36
10	severity=1, resources=3, initialTime=0, availability=0, lottery=8, fcfs=6, sickestFirst=1, youngestFirst=7, livesSaved=24, prognosis=24, instrumental=24, reciprocity=6, age=28, survivalPerctentage=34, expectedYears=17
11	severity=4, resources=3, initialTime=0, availability=0, lottery=21, fcfs=21, sickestFirst=23, youngestFirst=9, livesSaved=2, prognosis=2, instrumental=5, reciprocity=21, age=29, survivalPerctentage=87, expectedYears=43
12	severity=3, resources=3, initialTime=0, availability=0, lottery=6, fcfs=22, sickestFirst=15, youngestFirst=13, livesSaved=13, prognosis=13, instrumental=0, reciprocity=22, age=36, survivalPerctentage=65, expectedYears=32
13	severity=3, resources=2, initialTime=0, availability=0, lottery=20, fcfs=5, sickestFirst=17, youngestFirst=27, livesSaved=16, prognosis=16, instrumental=13, reciprocity=5, age=66, survivalPerctentage=56, expectedYears=28
14	severity=2, resources=2, initialTime=0, availability=0, lottery=26, fcfs=10, sickestFirst=8, youngestFirst=29, livesSaved=20, prognosis=20, instrumental=20, reciprocity=10, age=69, survivalPerctentage=41, expectedYears=20
15	severity=1, resources=2, initialTime=0, availability=0, lottery=13, fcfs=2, sickestFirst=2, youngestFirst=24, livesSaved=29, prognosis=29, instrumental=2, reciprocity=2, age=63, survivalPerctentage=12, expectedYears=6
16	severity=3, resources=3, initialTime=0, availability=0, lottery=2, fcfs=26, sickestFirst=16, youngestFirst=1, livesSaved=15, prognosis=15, instrumental=4, reciprocity=26, age=23, survivalPerctentage=59, expectedYears=29
17	severity=4, resources=2, initialTime=0, availability=0, lottery=29, fcfs=9, sickestFirst=24,

	youngestFirst=0, livesSaved=3, prognosis=3, instrumental=1, reciprocity=9, age=21, survivalPerctentage=78, expectedYears=39
18	severity=4, resources=1, initialTime=0, availability=0, lottery=18, fcfs=16, sickestFirst=29, youngestFirst=8, livesSaved=0, prognosis=0, instrumental=27, reciprocity=16, age=28, survivalPerctentage=97, expectedYears=48
19	severity=2, resources=2, initialTime=0, availability=0, lottery=9, fcfs=19, sickestFirst=9, youngestFirst=28, livesSaved=18, prognosis=18, instrumental=6, reciprocity=19, age=67, survivalPerctentage=51, expectedYears=25
20	severity=3, resources=1, initialTime=0, availability=0, lottery=1, fcfs=27, sickestFirst=22, youngestFirst=2, livesSaved=6, prognosis=6, instrumental=8, reciprocity=27, age=23, survivalPerctentage=75, expectedYears=37
21	severity=2, resources=1, initialTime=0, availability=0, lottery=17, fcfs=15, sickestFirst=11, youngestFirst=5, livesSaved=19, prognosis=19, instrumental=9, reciprocity=15, age=27, survivalPerctentage=42, expectedYears=21
22	severity=3, resources=3, initialTime=0, availability=0, lottery=22, fcfs=8, sickestFirst=13, youngestFirst=16, livesSaved=11, prognosis=11, instrumental=15, reciprocity=8, age=44, survivalPerctentage=69, expectedYears=34
23	severity=2, resources=2, initialTime=0, availability=0, lottery=7, fcfs=20, sickestFirst=10, youngestFirst=4, livesSaved=25, prognosis=25, instrumental=10, reciprocity=20, age=25, survivalPerctentage=33, expectedYears=16
24	severity=2, resources=1, initialTime=0, availability=0, lottery=15, fcfs=24, sickestFirst=12, youngestFirst=11, livesSaved=23, prognosis=23, instrumental=7, reciprocity=24, age=36, survivalPerctentage=36, expectedYears=18
25	severity=2, resources=3, initialTime=0, availability=0, lottery=4, fcfs=0, sickestFirst=5, youngestFirst=18, livesSaved=22, prognosis=22, instrumental=29, reciprocity=0, age=48, survivalPerctentage=38, expectedYears=19
26	severity=3, resources=3, initialTime=0, availability=0, lottery=27, fcfs=12, sickestFirst=14, youngestFirst=12, livesSaved=10, prognosis=10, instrumental=18, reciprocity=12, age=36, survivalPerctentage=71, expectedYears=35
27	severity=2, resources=2, initialTime=0, availability=0, lottery=16, fcfs=7, sickestFirst=7, youngestFirst=15, livesSaved=17, prognosis=17, instrumental=22, reciprocity=7, age=44,

	survivalPerctentage=52, expectedYears=26
28	severity=3, resources=2, initialTime=0, availability=0, lottery=28, fcfs=17, sickestFirst=18, youngestFirst=10, livesSaved=8, prognosis=8, instrumental=28, reciprocity=17, age=34, survivalPerctentage=72, expectedYears=36
29	severity=4, resources=1, initialTime=0, availability=0, lottery=5, fcfs=13, sickestFirst=28, youngestFirst=20, livesSaved=4, prognosis=4, instrumental=21, reciprocity=13, age=54, survivalPerctentage=78, expectedYears=39
30	severity=1, resources=3, initialTime=0, availability=0, lottery=23, fcfs=3, sickestFirst=0, youngestFirst=6, livesSaved=27, prognosis=27, instrumental=14, reciprocity=3, age=28, survivalPerctentage=18, expectedYears=9

Ambulance

ID	**Values**
34	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
35	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
36	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
37	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
38	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
39	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
40	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
41	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
42	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
43	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
44	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
45	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
46	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
47	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
48	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
49	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
50	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
51	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false
52	driver=1, initialTime=0, availability=0, longitude=0, latitude=0, driving=false

53	driver=2, initialTime=0, availability=0, longitude=0, latitude=0, driving=false

Hospital

ID	Values
31	resources=1, currentPatients=0, capacity=8, distance=2, longitude=3, latitude=-4
32	resources=2, currentPatients=0, capacity=12, distance=1, longitude=-3, latitude=0
33	resources=2, currentPatients=0, capacity=12, distance=2, longitude=1, latitude=4

Random restart = 10 – allocation value = 525

Patient ID	Ambulance ID	Hospital ID
1	36	33
2	44	31
3	45	31
4	Null Resource	Null Resource
5	34	31
6	Null Resource	Null Resource
7	37	32
8	50	32
9	Null Resource	Null Resource
10	53	32
11	Null Resource	Null Resource
12	Null Resource	Null Resource
13	27	32
14	35	32
15	38	32
16	Null Resource	Null Resource
17	Null Resource	Null Resource
18	Null Resource	Null Resource
19	39	32
20	51	31
21	52	31
22	Null Resource	Null Resource

23	40	32
24	43	31
25	47	32
26	Null Resource	Null Resource
27	48	32
28	41	32
29	42	31
30	49	32

Random restart = 1 – allocation value = 526

Patient ID	**Ambulance ID**	**Hospital ID**
1	43	32
2	51	31
3	44	32
4	Null Resource	Null Resource
5	40	31
6	48	33
7	36	31
8	35	32
9	Null Resource	Null Resource
10	37	33
11	Null Resource	Null Resource
12	Null Resource	Null Resource
13	50	32
14	38	33
15	39	32
16	Null Resource	Null Resource
17	Null Resource	Null Resource
18	Null Resource	Null Resource
19	46	32
20	45	31

21	47	32
22	Null Resource	Null Resource
23	52	32
24	49	31
25	42	32
26	Null Resource	Null Resource
27	34	32
28	41	32
29	Null Resource	Null Resource
30	53	32

Appendix F
Display Results of Local Search in Ambulance and Hospital Stages

This section provides some of the attributes that may be returned when displaying the results. Note that the allocations can also be displayed.

Display Results of Ambulance Stage

These values were generated from performing the local search using Utilitarian social welfare and no principle policy on Test 90 of the Ambulance stage with 30 patients, 20 ambulances, 3 hospitals, 100 random restarts, and at time step 0.

Time	0
#Patients Available	10
#Patients	30
#Ambulances Available	0
# Ambulances	20
# Ambulances Driving To Incident	0
# Current Patient in Hospital ID 31	7
# Capacity in Hospital ID 31	8
# Current Patient in Hospital ID 32	12
# Capacity in Hospital ID 32	12
# Current Patient in Hospital ID 33	1
# Capacity in Hospital ID 33	12
#Patients with Severity 1 Available	0
#Patients with Severity 1 Unavailable	5
#Patients with Severity 2 Available	0
#Patients with Severity 2 Unavailable	8
#Patients with Severity 3 Available	4
#Patients with Severity 3 Unavailable	6
#Patients with Severity 4 Available	6
#Patients with Severity 4 Unavailable	1
#Resources Not Used	0

#Patients without Resources	10
#Patients with Resources Not Met	11
Counter	44
Swaps	11
Changes	32
Final Time of Best Allocation	9398333
Total Time Taken	828445737
Tcost-Utiliarian	525
Tcost-Egalitarian	58
Tcost-Nash	2.96462086370774E+35

Display Results of Hospital Stage

These values were generated from performing the local search using Utilitarian social welfare and no principle policy on Test 0 of the Hospital stage with 10 patients, 6 resources, 100 random restarts, and at time step 0.

Time	0
#Patients Available	4
#Patients	10
#Resources Available	0
#Resources	6
#Patients with Severity 1 Available	0
#Patients with Severity 1 Unavailable	2
#Patients with Severity 2 Available	0
#Patients with Severity 2 Unavailable	3
#Patients with Severity 3 Available	0
#Patients with Severity 3 Unavailable	1
#Patients with Severity 4 Available	4
#Patients with Severity 4 Unavailable	0
#Resources Not Used	0
#Patients without Resources	4
#Patients with Resources Not Met	5

Counter	12
Swaps	6
Changes	5
Final Time of Best Allocation	5803736
Total Time Taken	68876560
Tcost-Utiliarian	289
Tcost-Egalitarian	44
Tcost-Nash	178102046453768

Appendix G
Extra Validation

This appendix displays additional output that was produced for both the Ambulance and Hospital stages.

G.1 Ambulance Stage

Initial Utilitarian Allocation of Resource Categorized by Severity

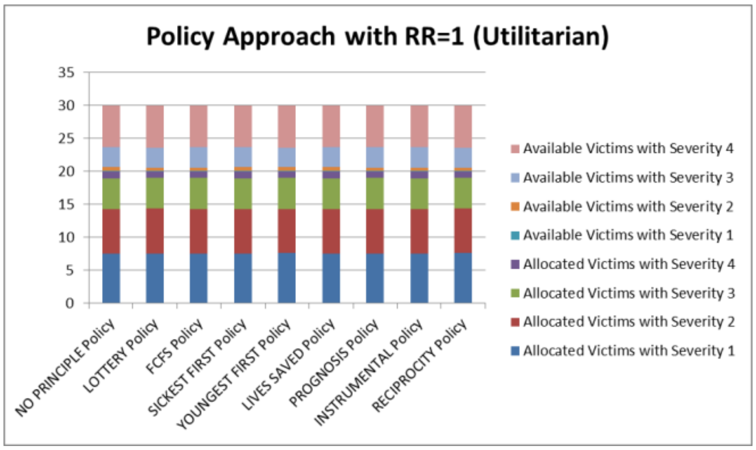

Figure VA-2b – Ambulance: Initial Allocation (Utilitarian Policy Approach with RR=1)

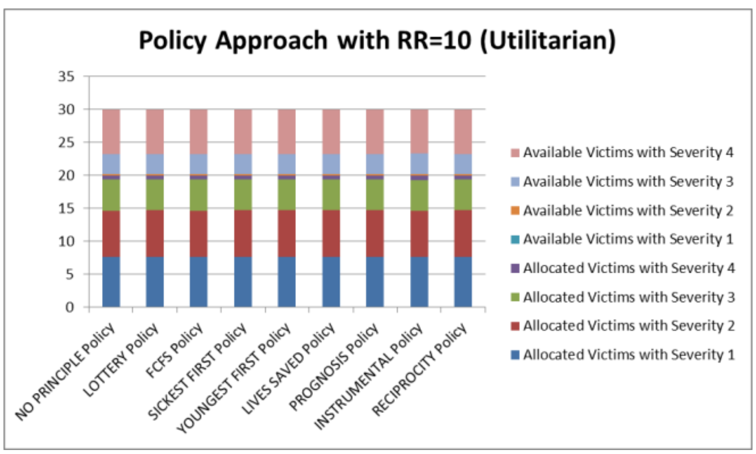

Figure VA-2c – Ambulance: Initial Allocation (Utilitarian Policy Approach with RR=10)

Initial Egalitarian Allocation Values

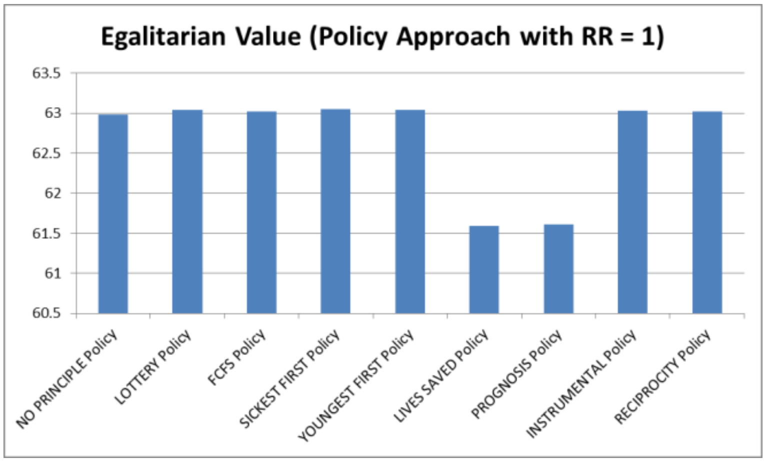

Figure VA-3b – Ambulance: Initial Egalitarian Allocation Value Policy Approach with RR=1

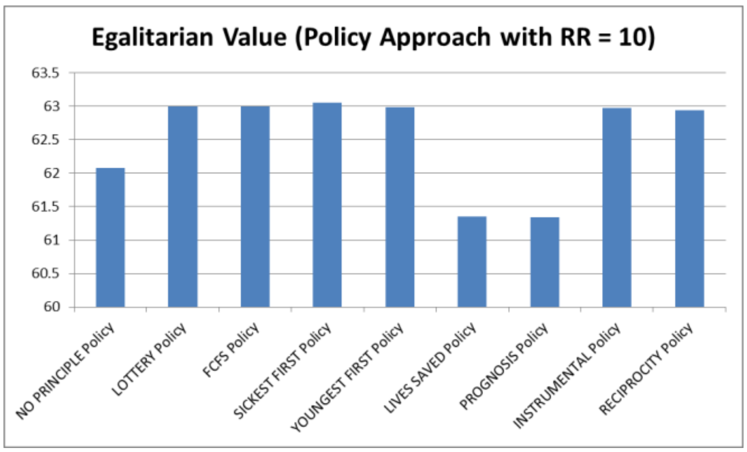

Figure VA-3c – Ambulance: Initial Egalitarian Allocation Value Policy Approach with RR=10

Initial Egalitarian Allocation of Resource Categorized by Severity

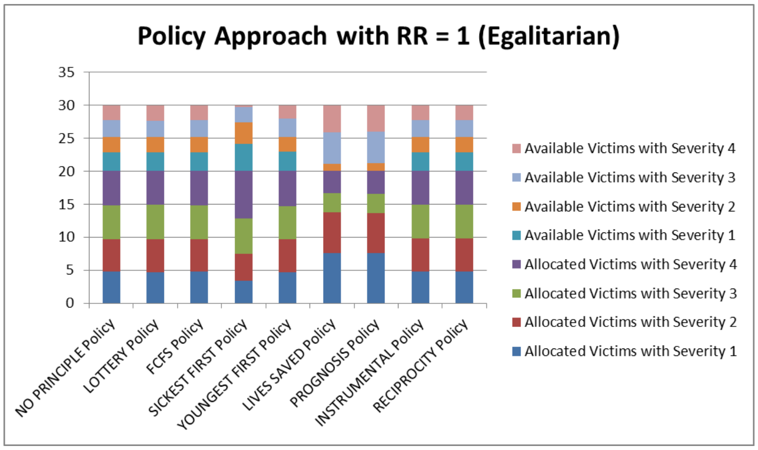

Figure VA-4b – Ambulance: Initial Allocation (Egalitarian Policy Approach with RR=1)

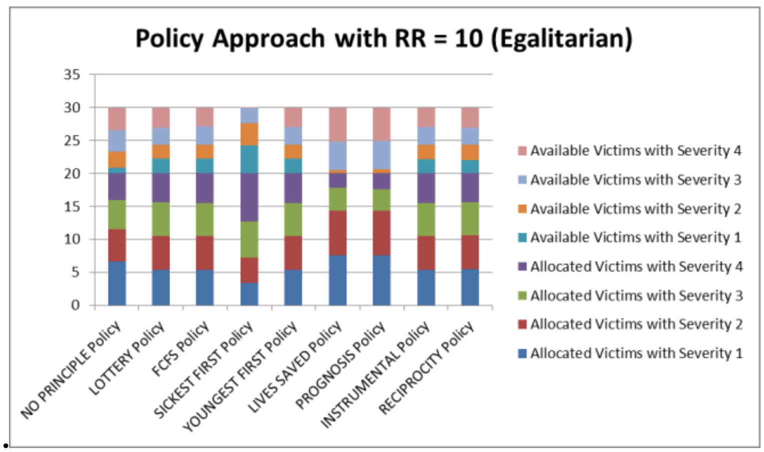

Figure VA-4c – Ambulance: Initial Allocation (Egalitarian Policy Approach with RR=10)

Initial Nash Product Allocation Values

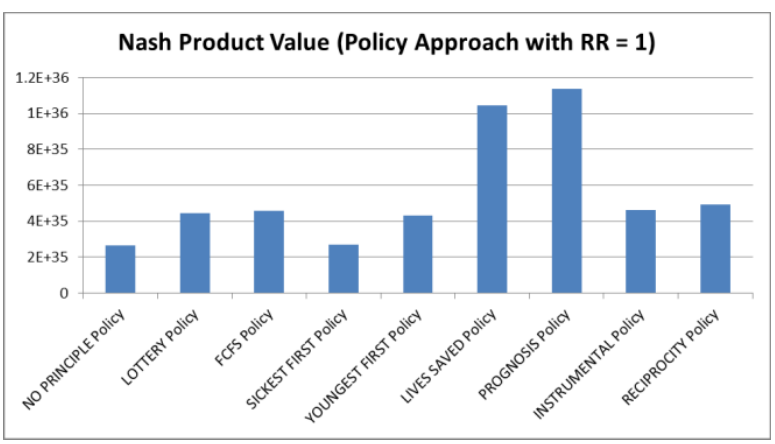

Figure VA-5b – Ambulance: Initial Nash Product Allocation Value Policy Approach with RR=1

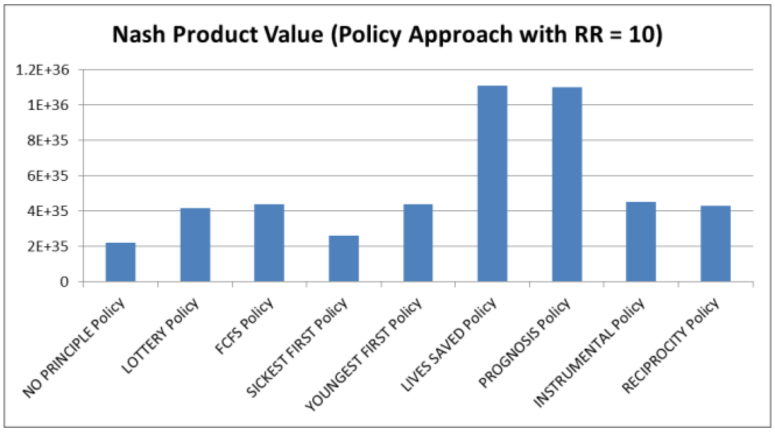

Figure VA-5c – Ambulance: Initial Nash Product Allocation Value Policy Approach with RR=10

Initial Nash Product Allocation of Resource Categorized by Severity

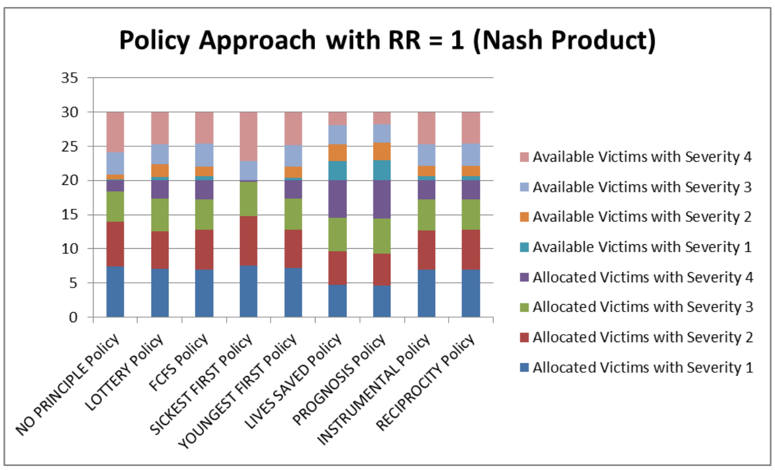

Figure VA-6b – Ambulance: Initial Allocation (Nash Product Policy Approach with RR=1)

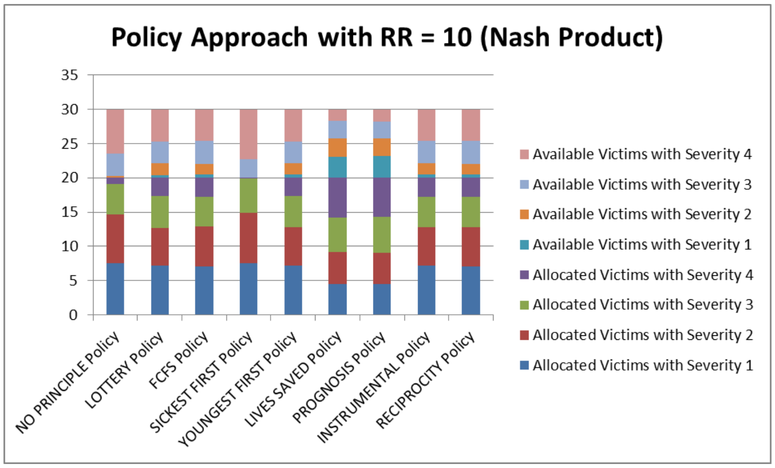

Figure VA-6c – Ambulance: Initial Allocation (Nash Product Policy Approach with RR=10)

Utilitarian Social Welfare over Time (No buffer time and sensing every time step)

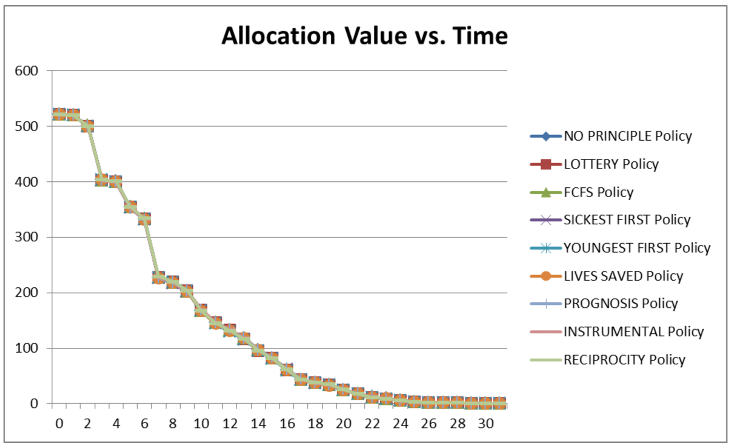

Figure VA-7a – Ambulance: Allocation Value over Time (Policy Approach)

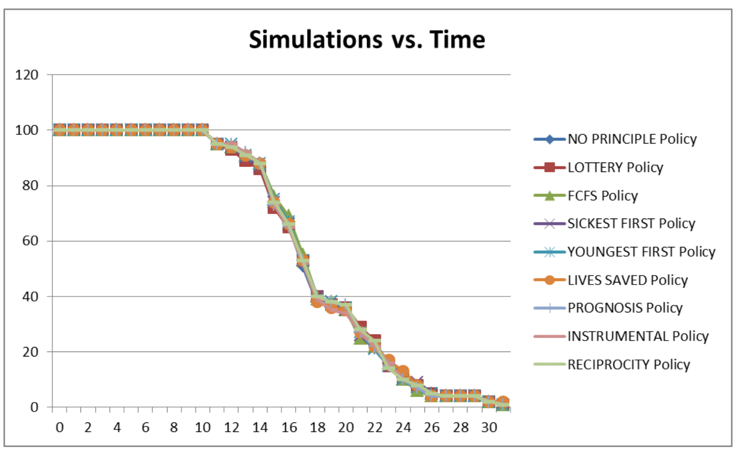

Figure VA-8a – Ambulance: Simulations over Time (Policy Approach)

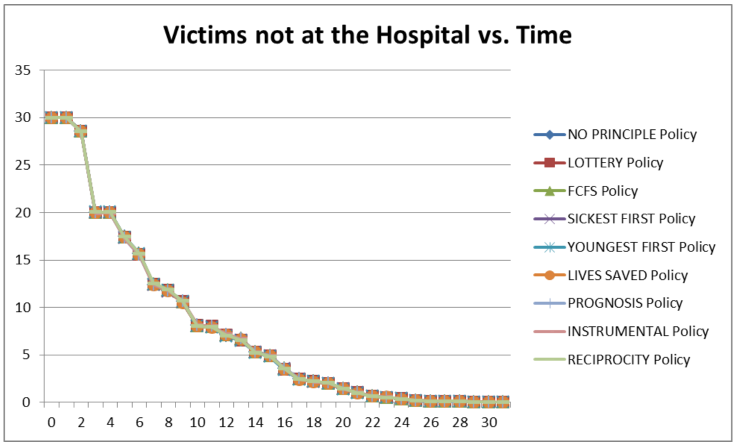

Figure VA-9a – Ambulance: Number of Victims not at the Hospital over Time (Policy Approach)

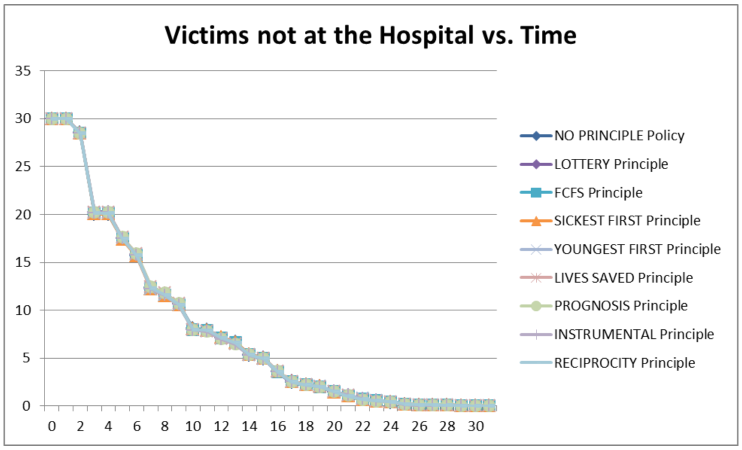

Figure VA-9b – Ambulance: Number of Victims not at the Hospital over Time (Principle Approach)

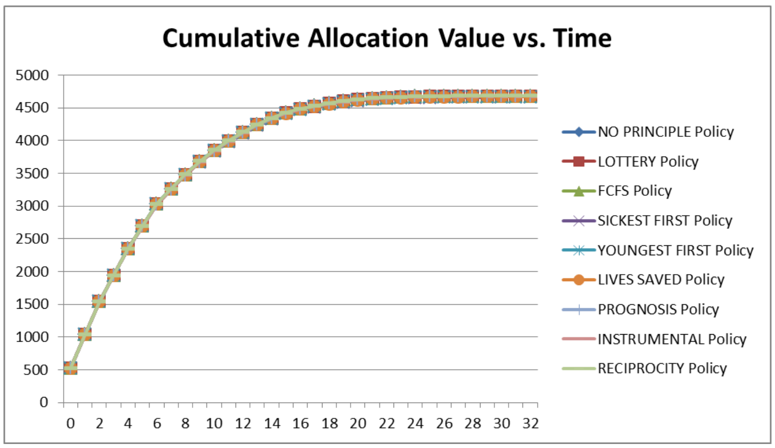

Figure VA-10a – Ambulance: Cumulative Allocation Value over Time (Policy Approach)

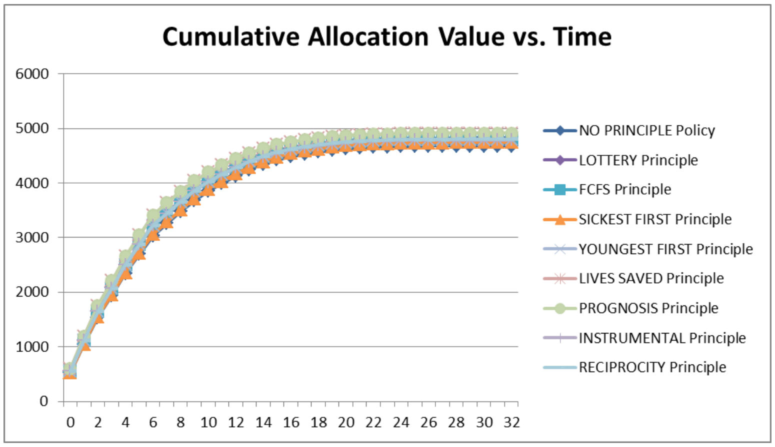

Figure VA-10b – Ambulance: Cumulative Allocation Value over Time (Principle Approach)

Utilitarian Social Welfare over Time (No buffer time and sensing every 2nd and 3rd time step)

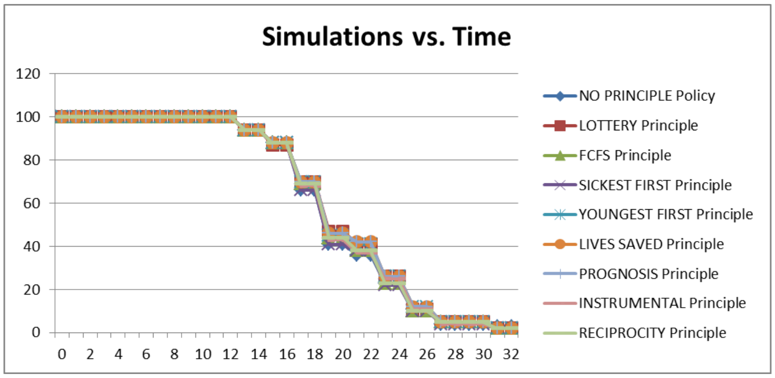

Figure VA-12a – Ambulance: Simulations over Time (Sensing=2)

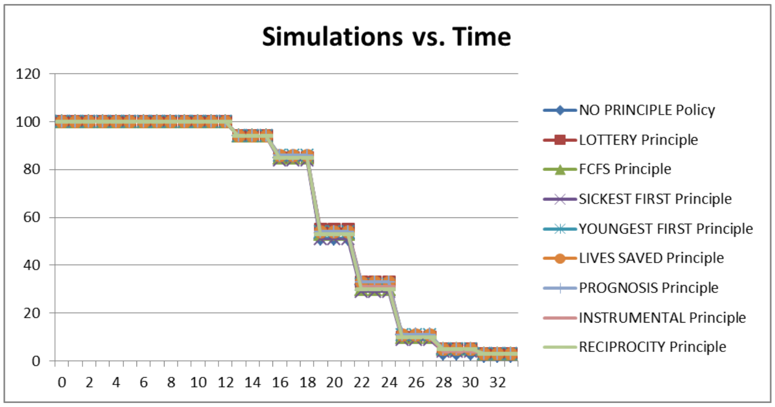

Figure VA-12b – Ambulance: Simulations over Time (Sensing=3)

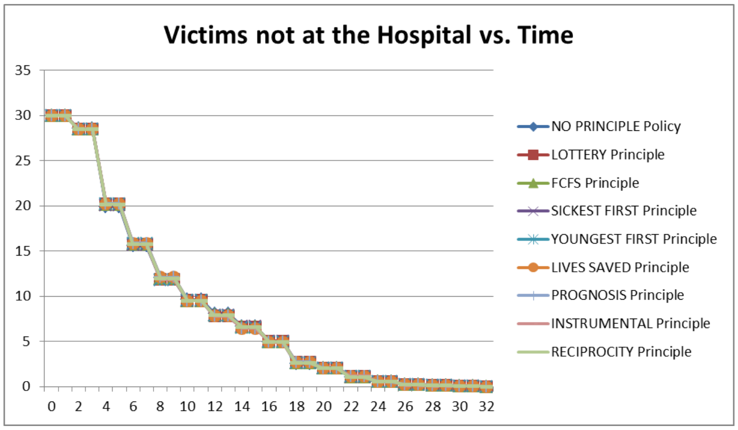

Figure VA-13a – Ambulance: Number of Victims not at the Hospital over Time (Principle Approach) (Sensing=2)

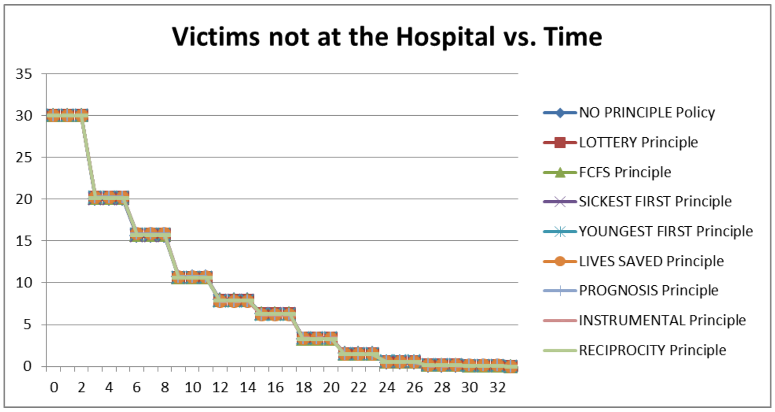

Figure VA-13b – Ambulance: Number of Victims not at the Hospital over Time (Principle Approach) (Sensing=3)

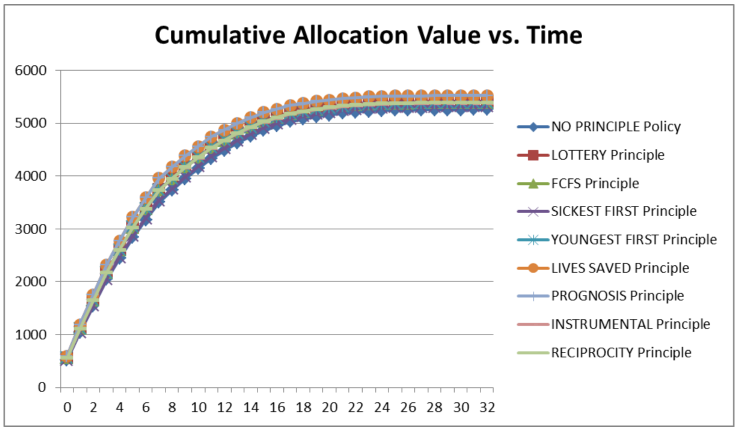

Figure VA-14a – Ambulance: Cumulative Allocation Value over Time (Principle Approach) (Sensing=2)

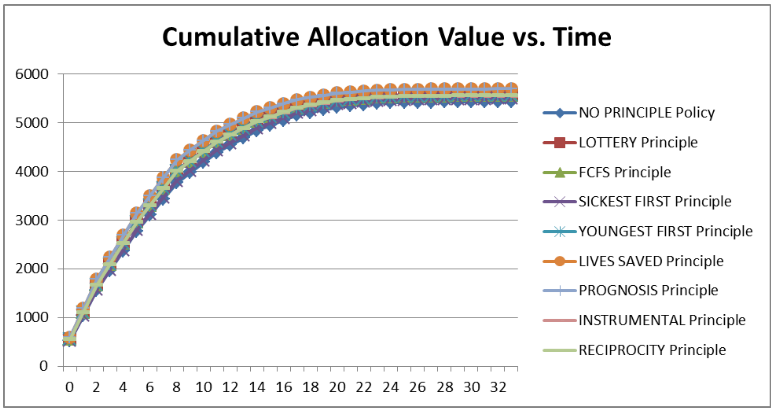

Figure VA-14b – Ambulance: Cumulative Allocation Value over Time (Principle Approach) (Sensing=3)

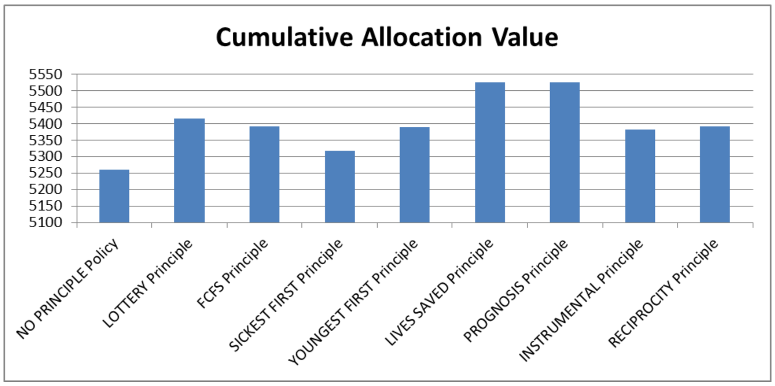

Figure VA-14c – Ambulance: Cumulative Allocation Value (Sensing=2)

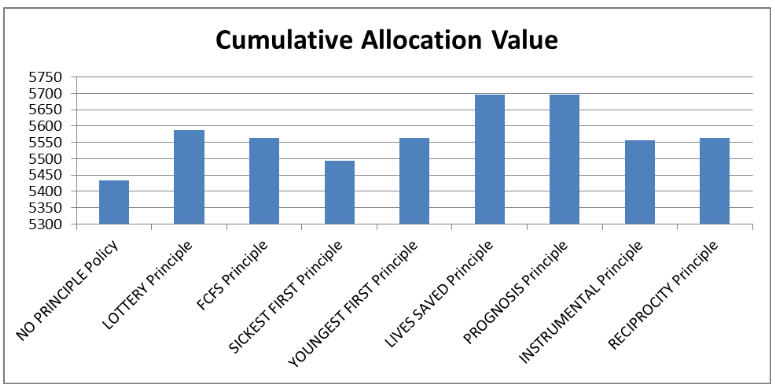

Figure VA-14d – Ambulance: Cumulative Allocation Value (Sensing=3)

Utilitarian Social Welfare over Time (Buffer time is 2 time steps and sensing every 1st, 2nd, and 3rd time step)

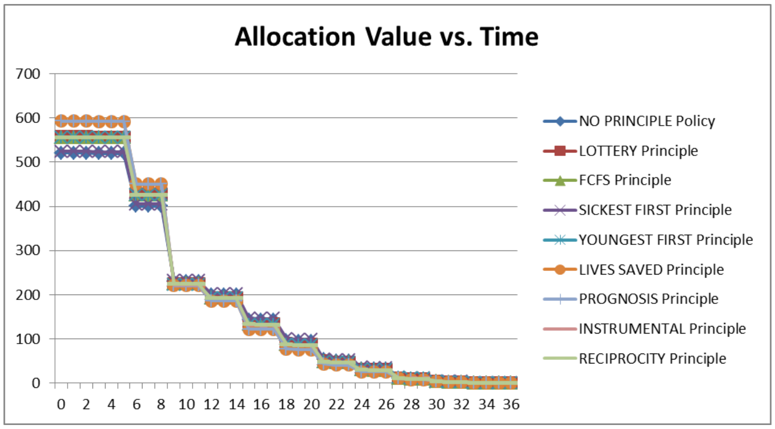

Figure VA-15c – Ambulance: Allocation Value over Time (Sensing=3, Buffer=2)

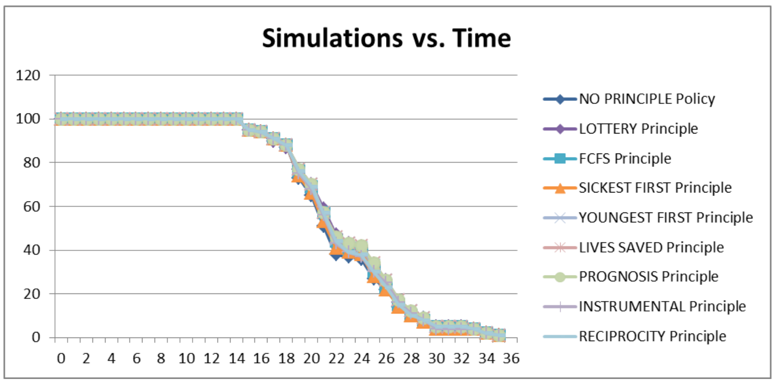

Figure VA-16a – Ambulance: Simulations over Time (Sensing=1, Buffer=2)

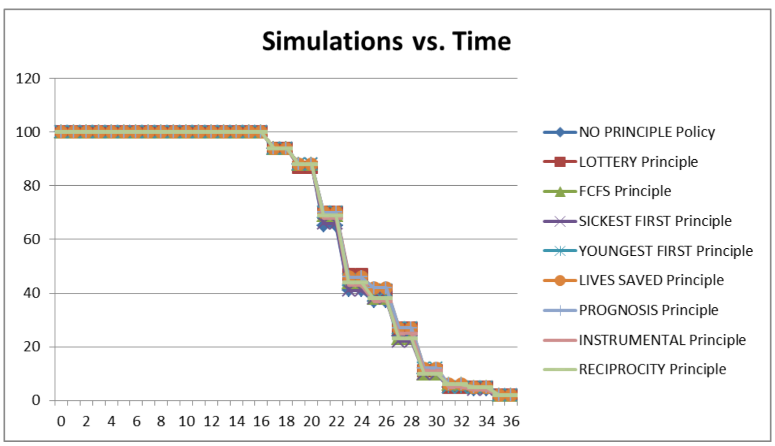

Figure VA-16b – Ambulance: Simulations over Time (Sensing=2, Buffer=2)

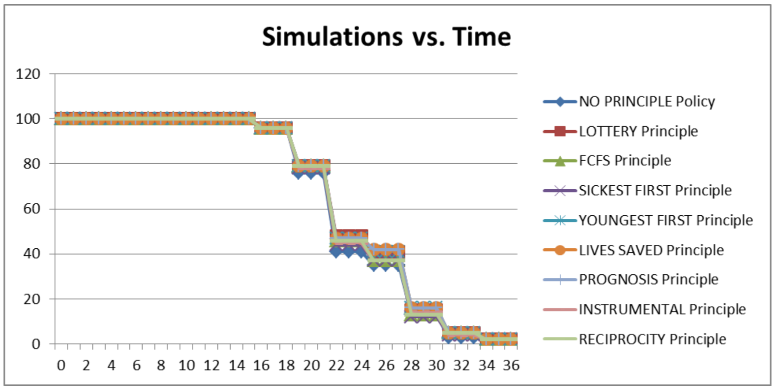

Figure VA-16c – Ambulance: Simulations over Time (Sensing=3, Buffer=2)

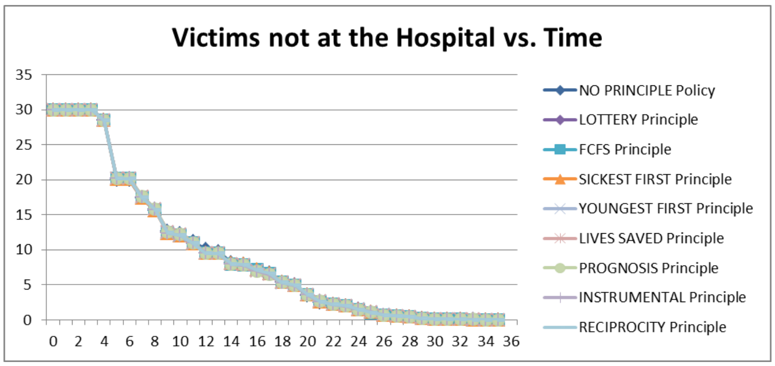

Figure VA-17a – Ambulance: Number of Victims not at the Hospital over Time (Sensing=1, Buffer=2)

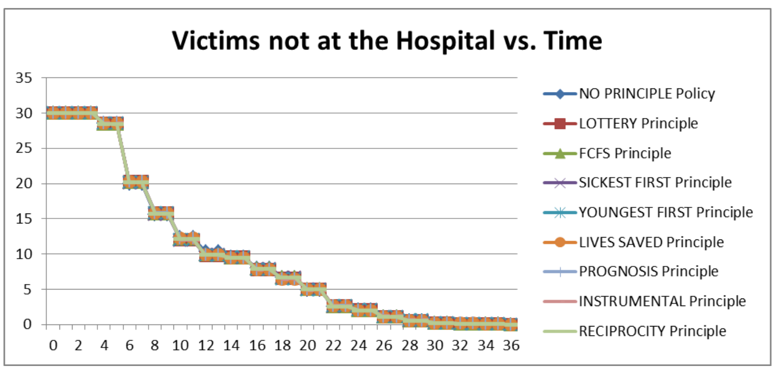

Figure VA-17b – Ambulance: Number of Victims not at the Hospital over Time (Sensing=2, Buffer=2)

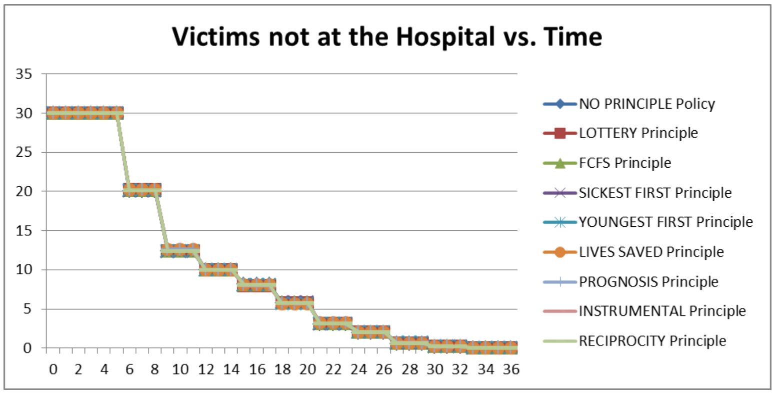

Figure VA-17c – Ambulance: Number of Victims not at the Hospital over Time (Sensing=3, Buffer=2)

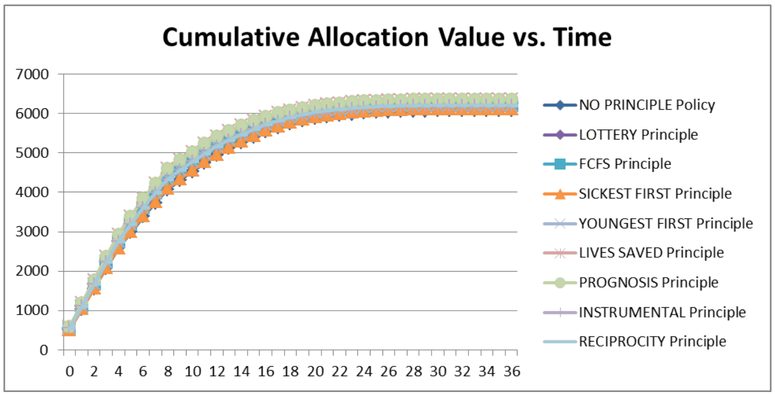

Figure VA-18a – Ambulance: Cumulative Allocation Value over Time (Sensing=1, Buffer=2)

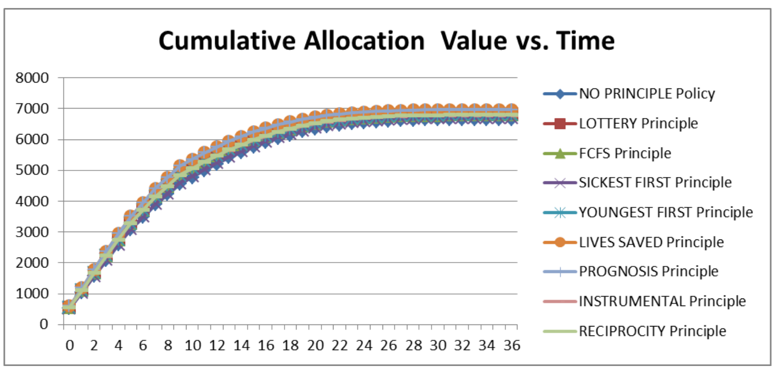

Figure VA-18b – Ambulance: Cumulative Allocation Value over Time (Sensing=2, Buffer=2)

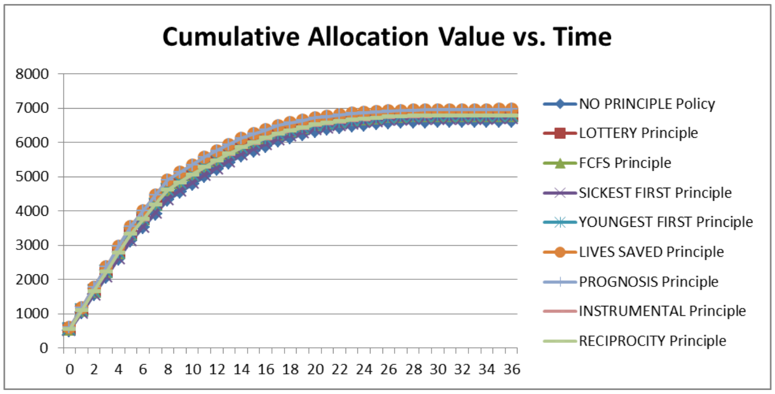

Figure VA-18c – Ambulance: Cumulative Allocation Value over Time (Sensing=3, Buffer=2)

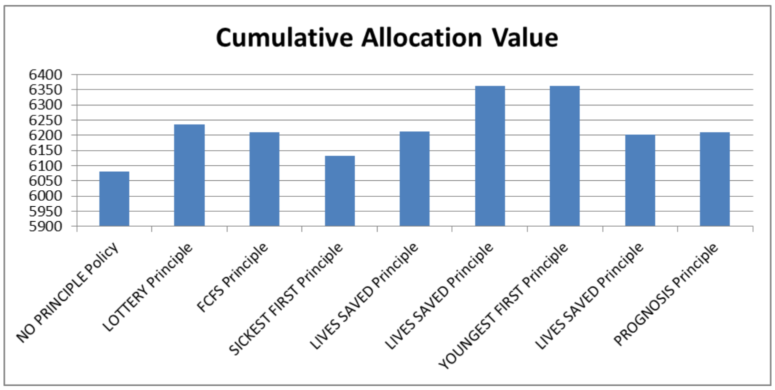

Figure VA-18d – Ambulance: Cumulative Allocation Value (Sensing=1, Buffer=2)

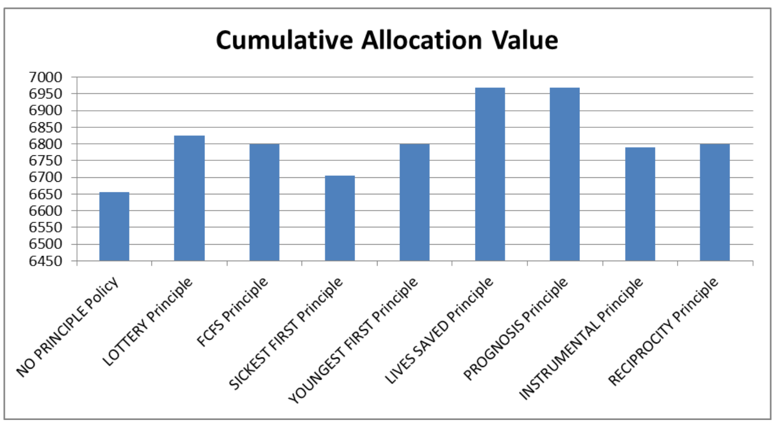

Figure VA-18e – Ambulance: Cumulative Allocation Value (Sensing=2, Buffer=2)

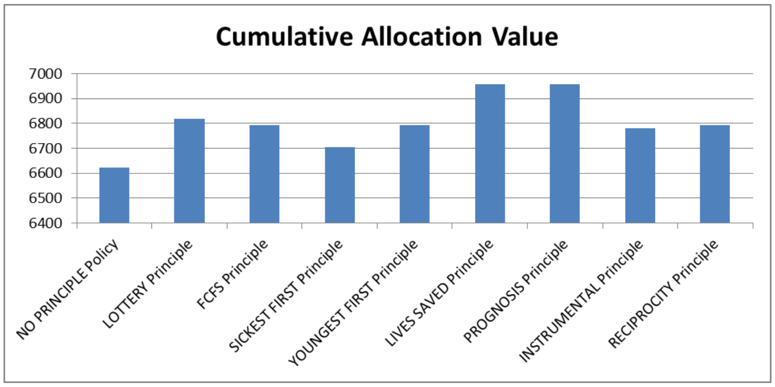

Figure VA-18f – Ambulance: Cumulative Allocation Value (Sensing=3, Buffer=2)

G.2 Hospital Stage

Initial Utilitarian Allocation Values

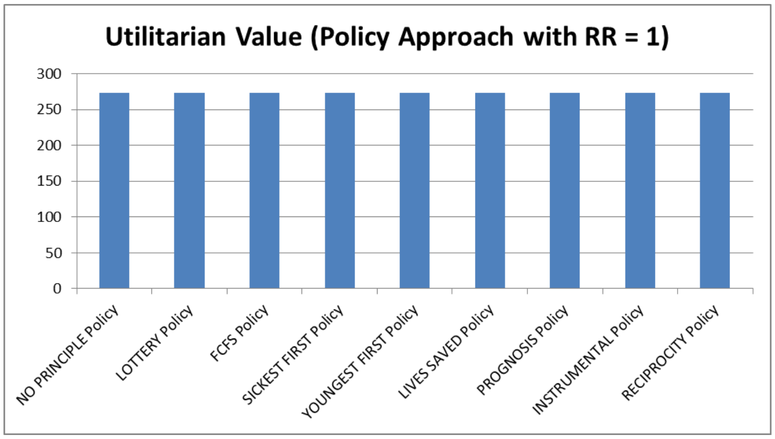

Figure VH-1b – Hospital: Initial Utilitarian Allocation Value Policy Approach with RR=1

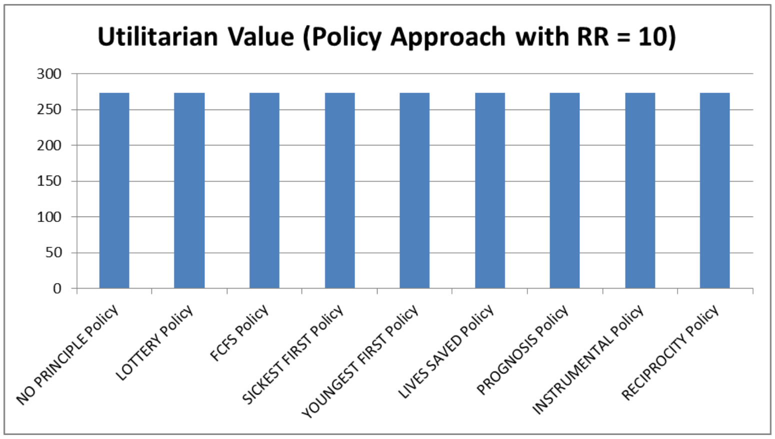

Figure VH-1c – Hospital: Initial Utilitarian Allocation Value Policy Approach with RR=10

Initial Utilitarian Allocation of Resource Categorized by Severity

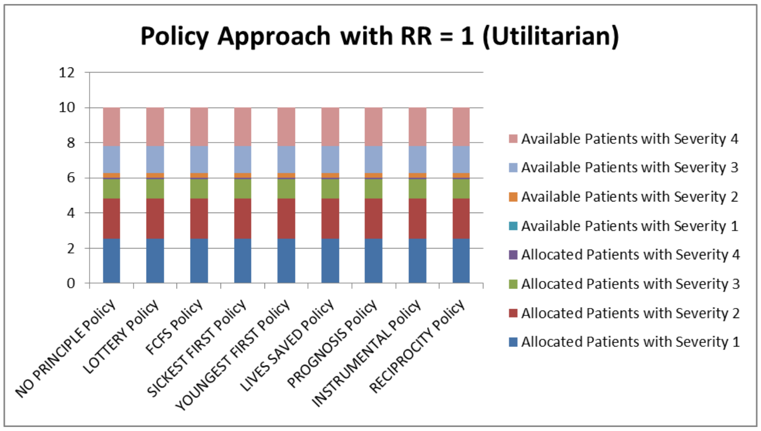

Figure VH-2b – Hospital: Initial Allocation (Utilitarian Policy Approach with RR=1)

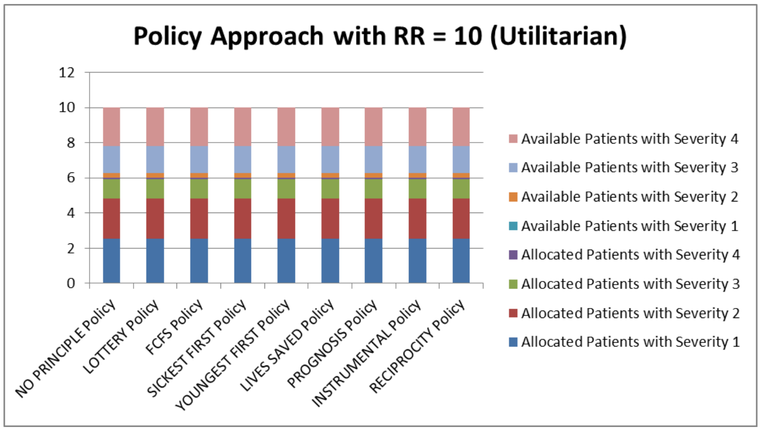

Figure VH-2c – Hospital: Initial Allocation (Utilitarian Policy Approach with RR=10)

Initial Egalitarian Allocation Values

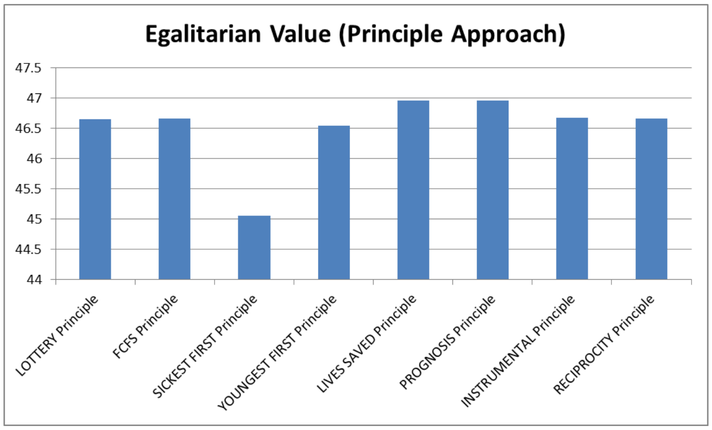

Figure VH-3a – Hospital: Initial Egalitarian Allocation Value Principle Approach

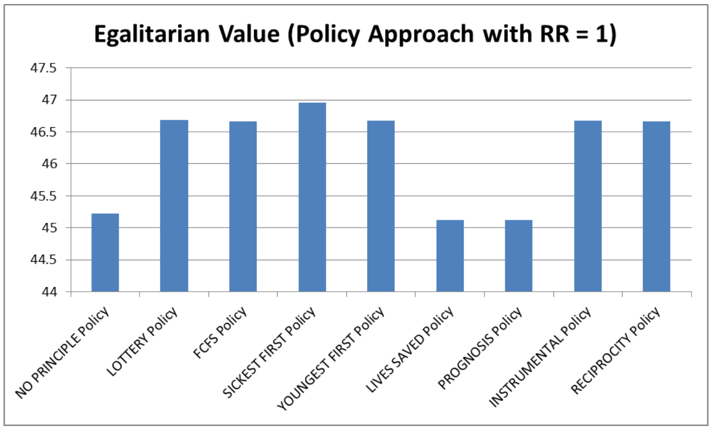

Figure VH-3b – Hospital: Initial Egalitarian Allocation Value Policy Approach with RR=1

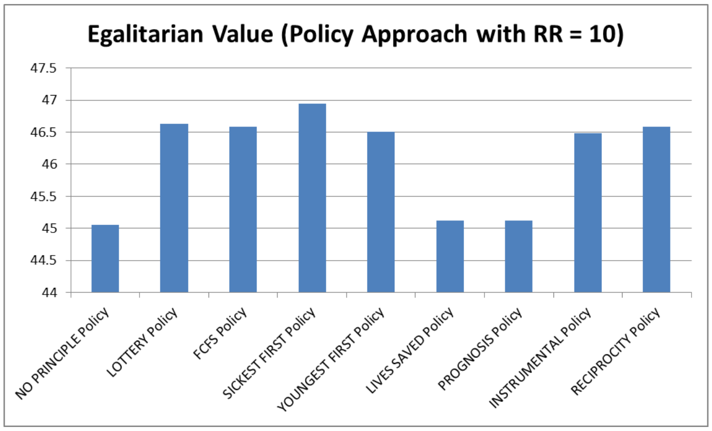

Figure VH-3c – Hospital: Initial Egalitarian Allocation Value Policy Approach with RR=10

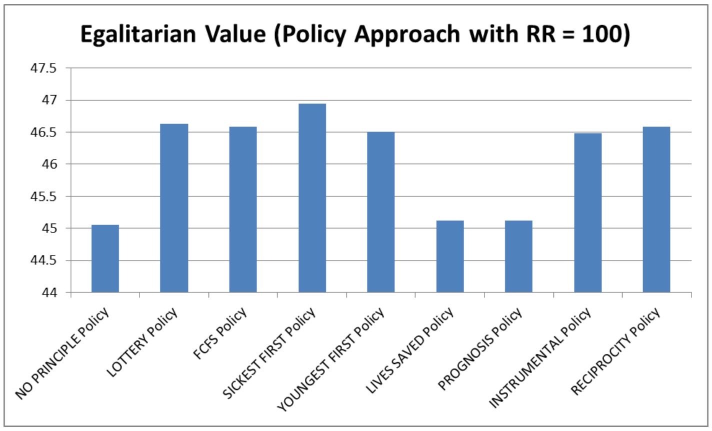

Figure VH-3d – Hospital: Initial Egalitarian Allocation Value Policy Approach with RR=100

Initial Egalitarian Allocation of Resource Categorized by Severity

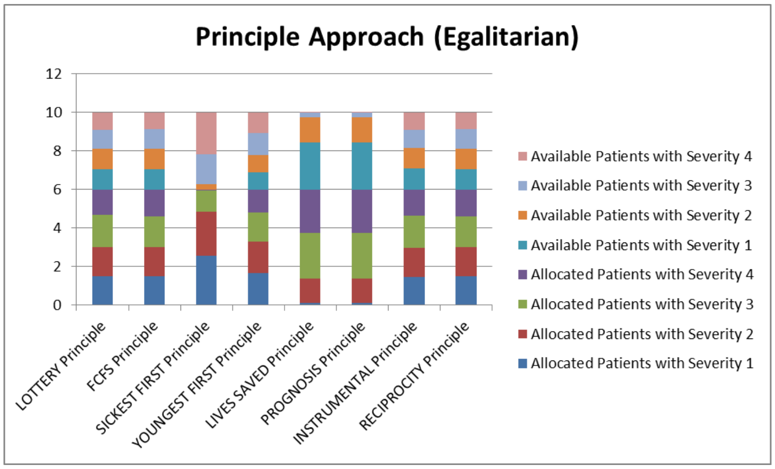

Figure VH-4a – Hospital: Initial Allocation (Egalitarian Principle Approach)

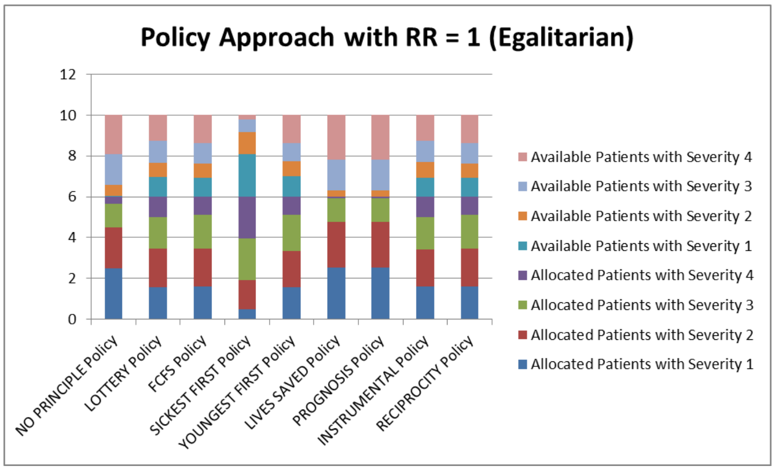

Figure VH-4b – Hospital: Initial Allocation (Egalitarian Policy Approach with RR=1)

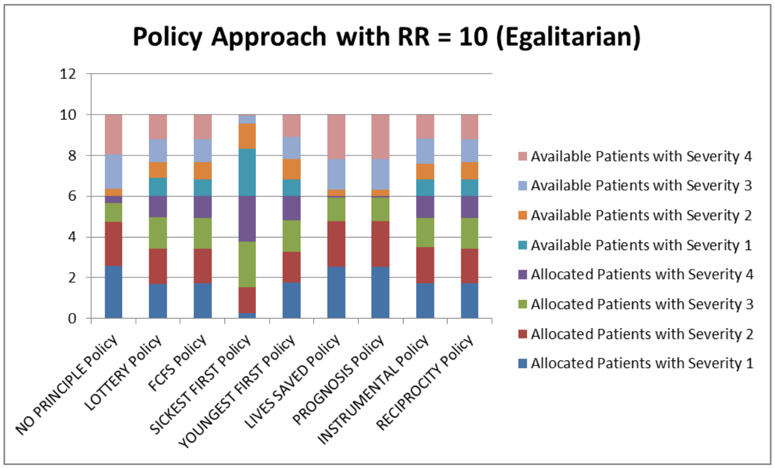

Figure VH-4c – Hospital: Initial Allocation (Egalitarian Policy Approach with RR=10)

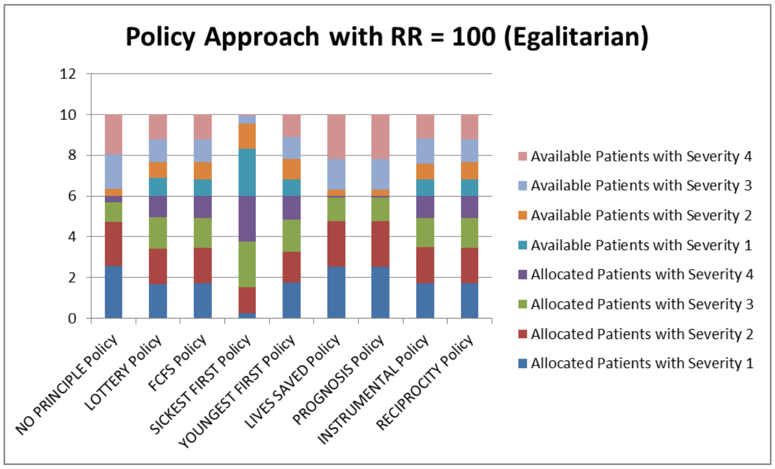

Figure VH-4d – Hospital: Initial Allocation (Egalitarian Policy Approach with RR=100)

Initial Nash Product Allocation Values

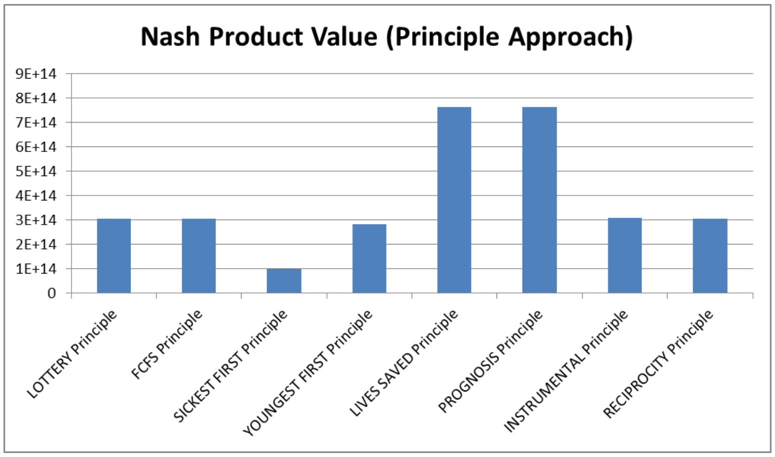

Figure VH-5a – Hospital: Initial Nash Product Allocation Value Principle Approach

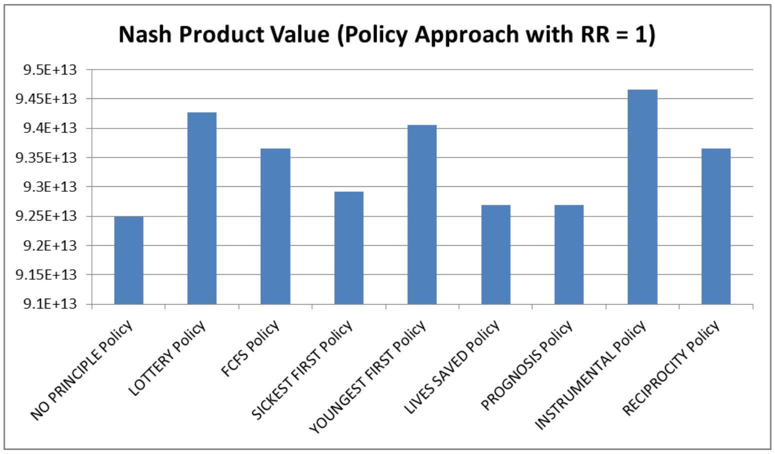

Figure VH-5b – Hospital: Initial Nash Product Allocation Value Policy Approach with RR=1

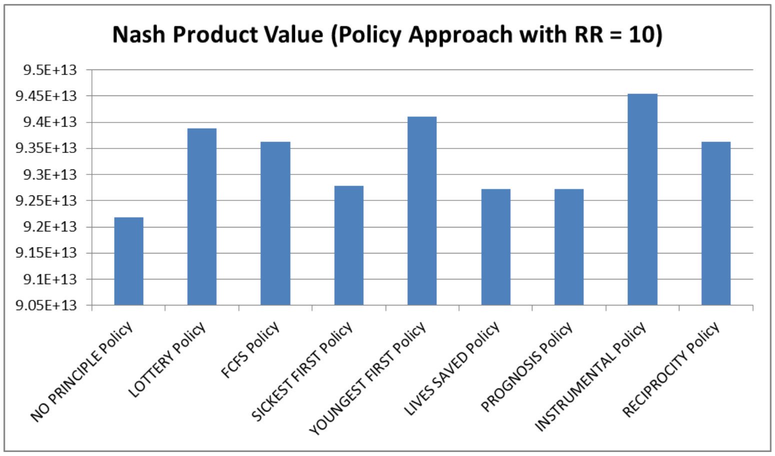

Figure VH-5c – Hospital: Initial Nash Product Allocation Value Policy Approach with RR=10

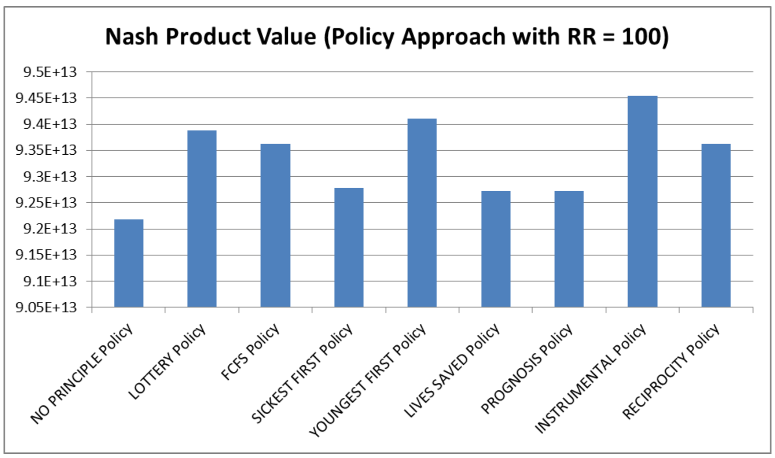

Figure VH-5d – Hospital: Initial Nash Product Allocation Value Policy Approach with RR=100

Initial Nash Product Allocation of Resource Categorized by Severity

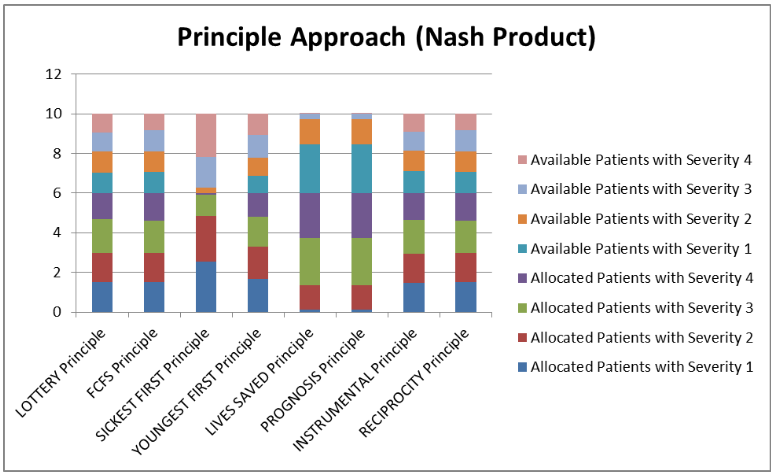

Figure VH-6a – Hospital: Initial Allocation (Nash Product Principle Approach)

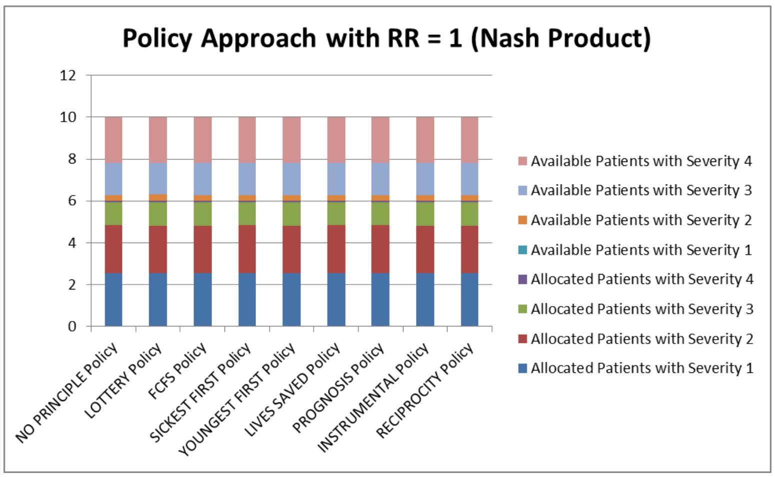

Figure VH-6b – Hospital: Initial Allocation (Nash Product Policy Approach with R=1)

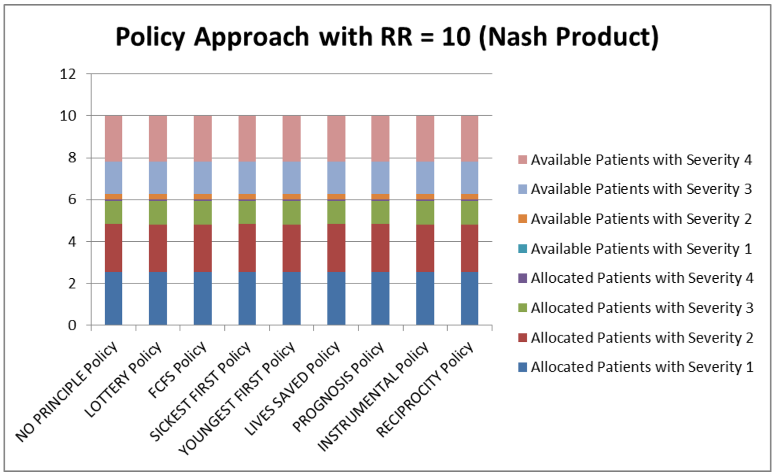

Figure VH-6c – Hospital: Initial Allocation (Nash Product Policy Approach with R=10)

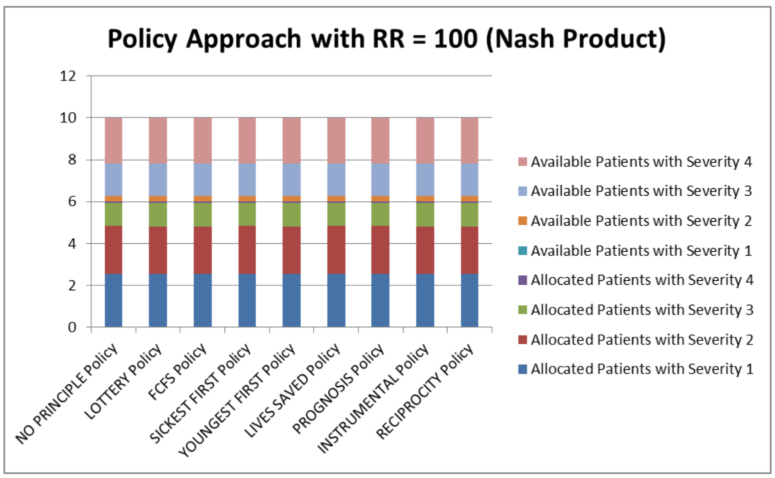

Figure VH-6d – Hospital: Initial Allocation (Nash Product Policy Approach with R=100)

Utilitarian Social Welfare over Time (No buffer time and sensing every time step)

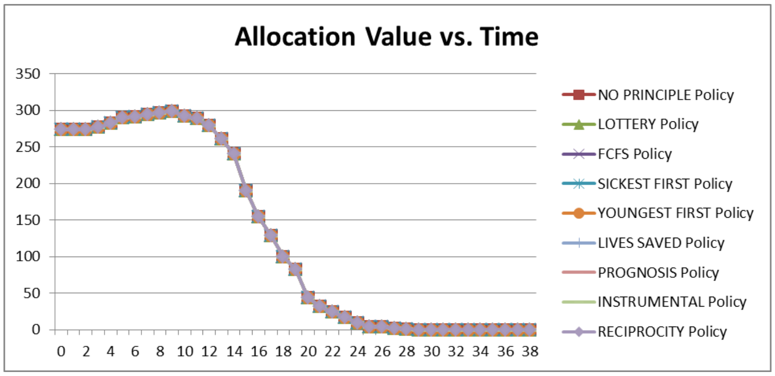

Figure VH-7a – Hospital: Allocation Value over Time (Policy Approach)

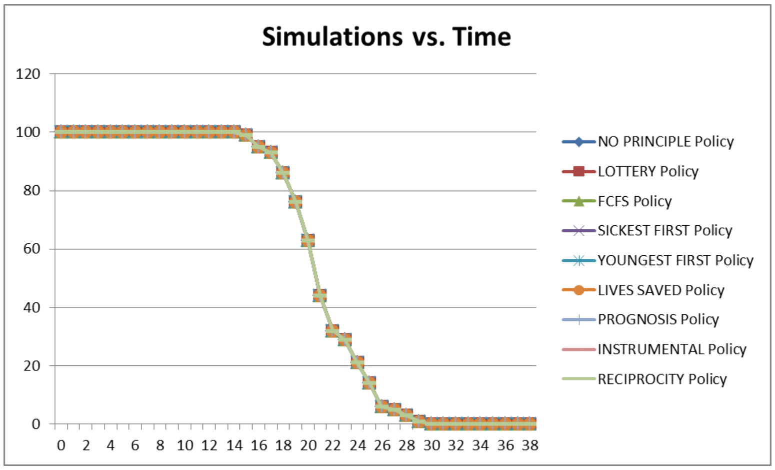

Figure VH-8a – Hospital: Simulations over Time (Policy Approach)

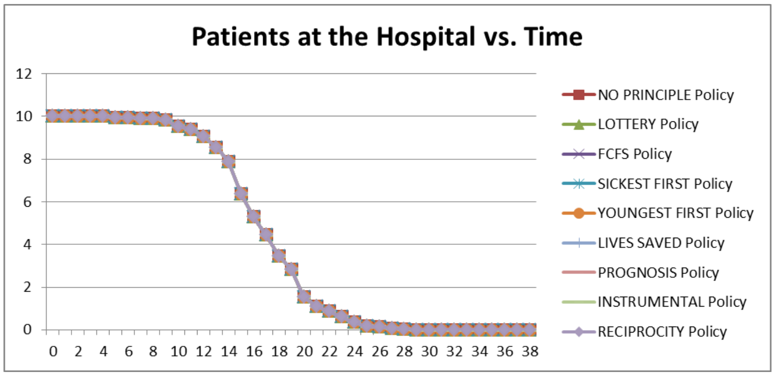

Figure VH-9a – Hospital: Number of Patient at the Hospital over Time (Policy Approach)

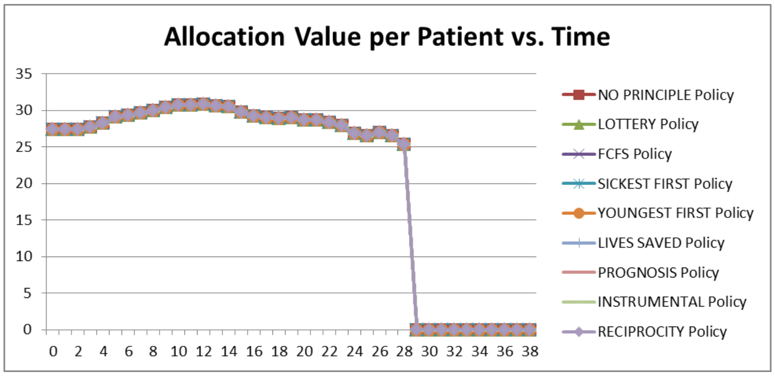

Figure VH-10a – Hospital: Allocation Value per Patient over Time (Policy Approach)

Utilitarian Social Welfare over Time (No buffer time and sensing every 2nd and 3rd time step)

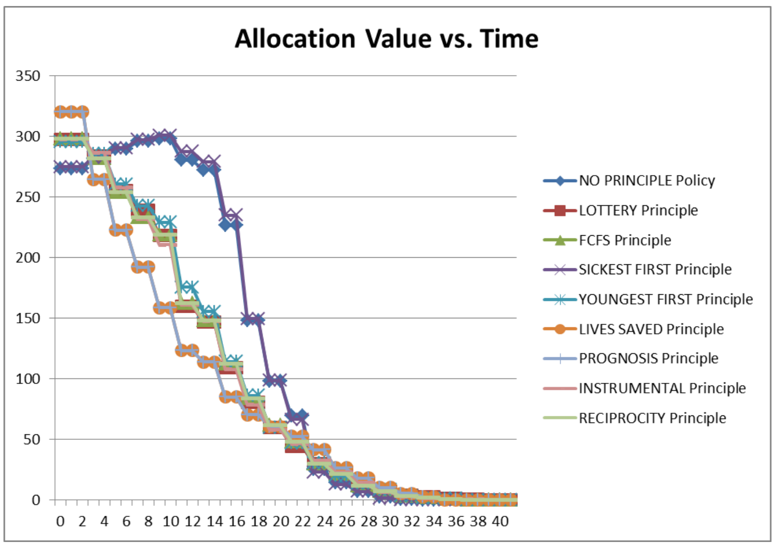

Figure VH-11a – Hospital: Allocation Value over Time (Sensing=2)

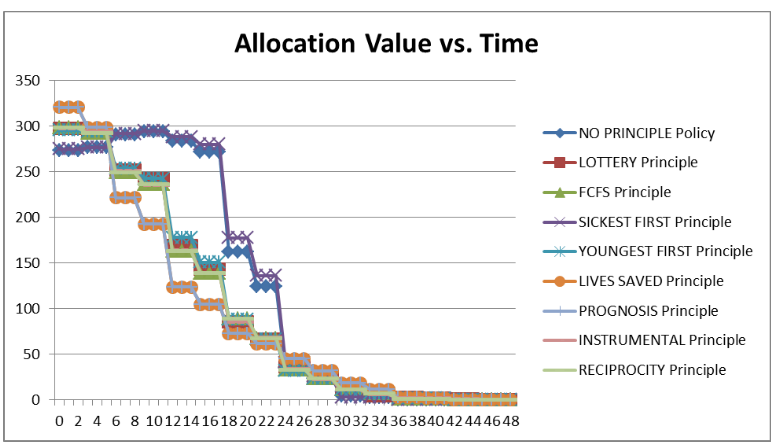

Figure VH-11b – Hospital: Allocation Value over Time (Sensing=3)

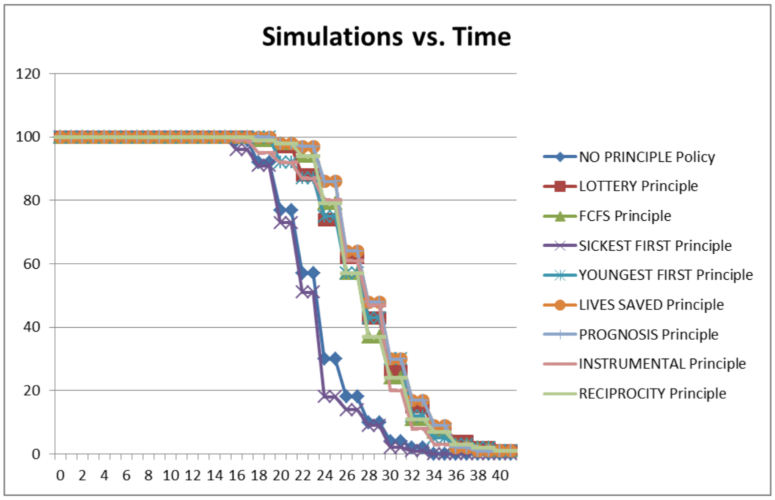

Figure VH-12a – Hospital: Simulations over Time (Sensing=2)

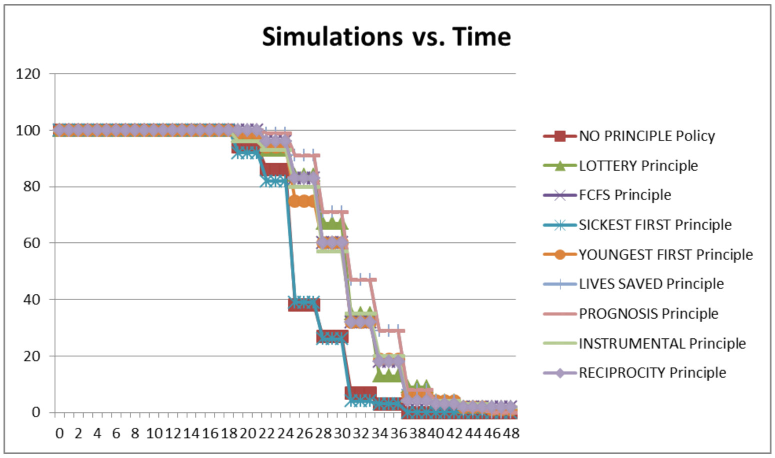

Figure VH-12b – Hospital: Simulations over Time (Sensing=3)

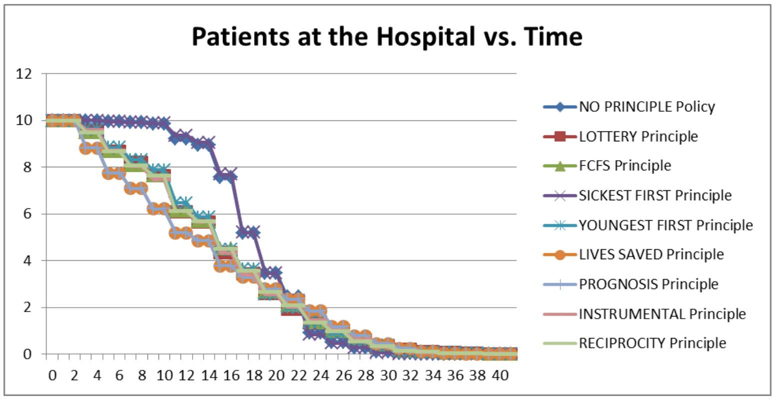

Figure VH-13a – Hospital: Number of Patient at the Hospital over Time (Sensing=2)

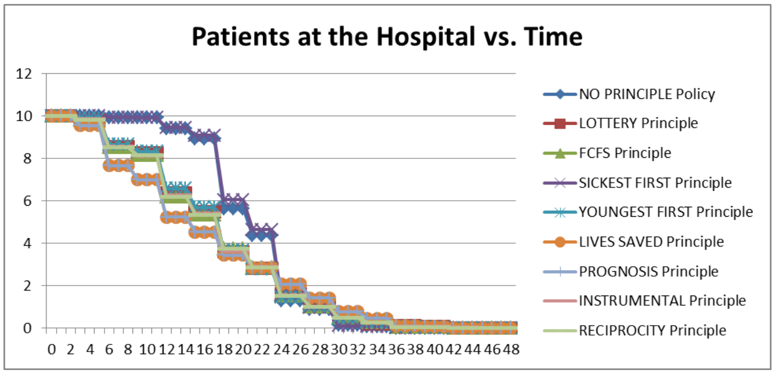

Figure VH-13b – Hospital: Number of Patient at the Hospital over Time (Sensing=3)

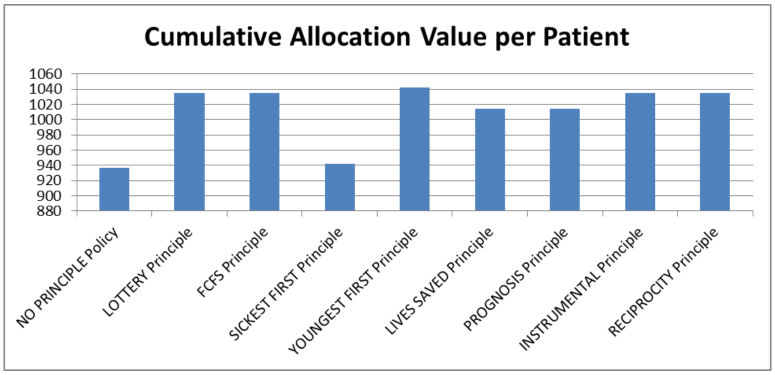

Figure VH-14c – Hospital: Number of Patient at the Hospital over Time (Sensing=2)

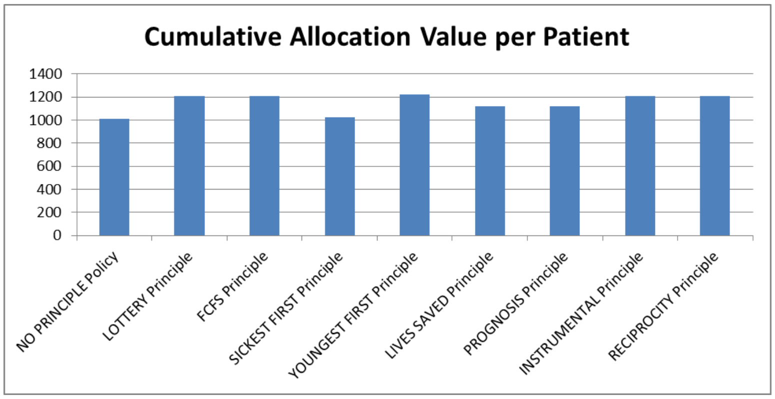

Figure VH-14d – Hospital: Number of Patient at the Hospital over Time (Sensing=3)

Utilitarian Social Welfare over Time (Buffer time is 1 time steps and sensing every 1st, 2nd, and 3rd time step)

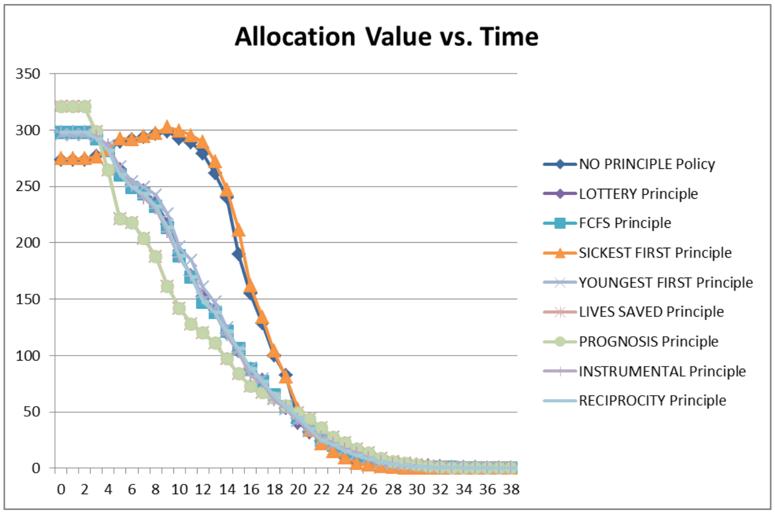

Figure VH-15a – Hospital: Allocation Value over Time (Sensing=1, Buffer=1)

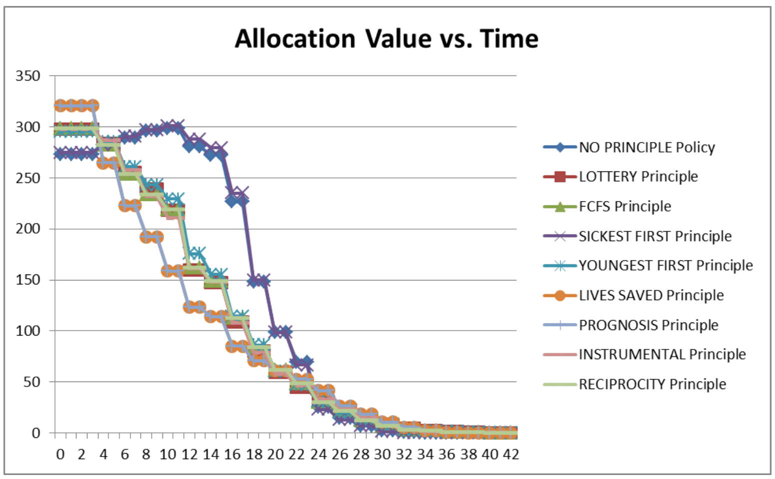

Figure VH-15b – Hospital: Allocation Value over Time (Sensing=2, Buffer=1)

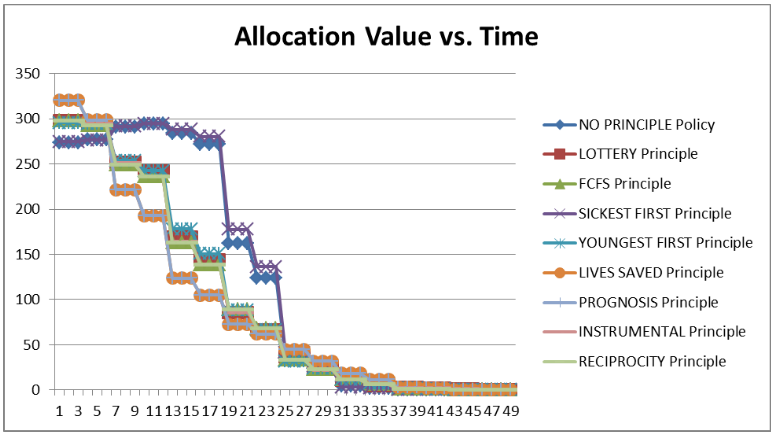

Figure VH-15c – Hospital: Allocation Value over Time (Sensing=3, Buffer=1)

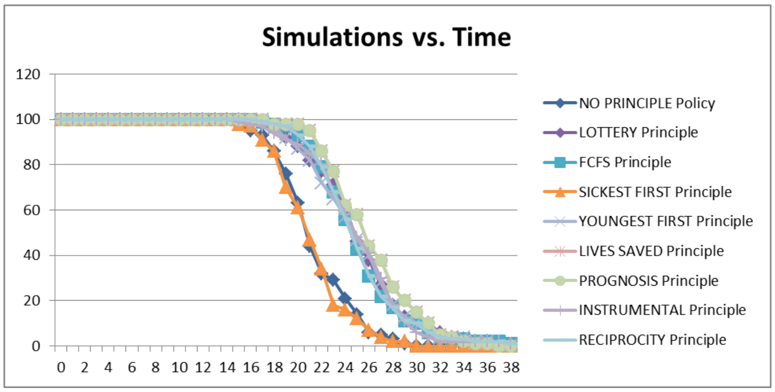

Figure VH-16a – Hospital: Simulations over Time (Sensing=1, Buffer=1)

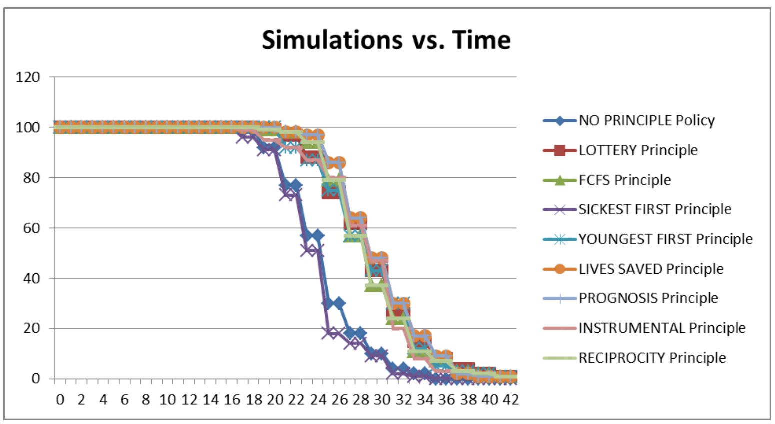

Figure VH-16b – Hospital: Simulations over Time (Sensing=2, Buffer=1)

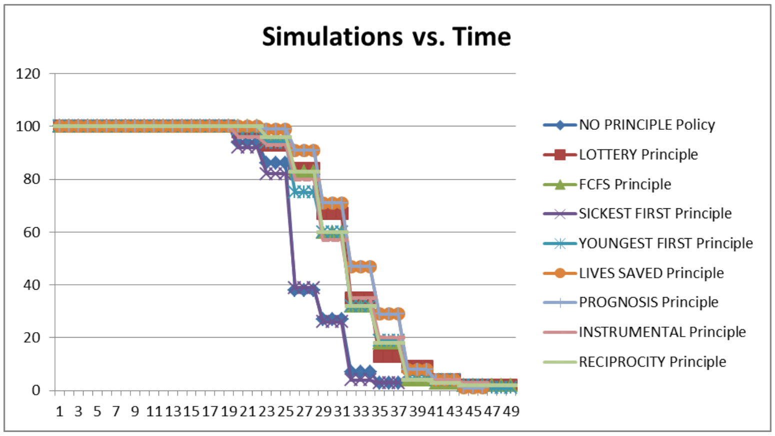

Figure VH-16c – Hospital: Simulations over Time (Sensing=3, Buffer=1)

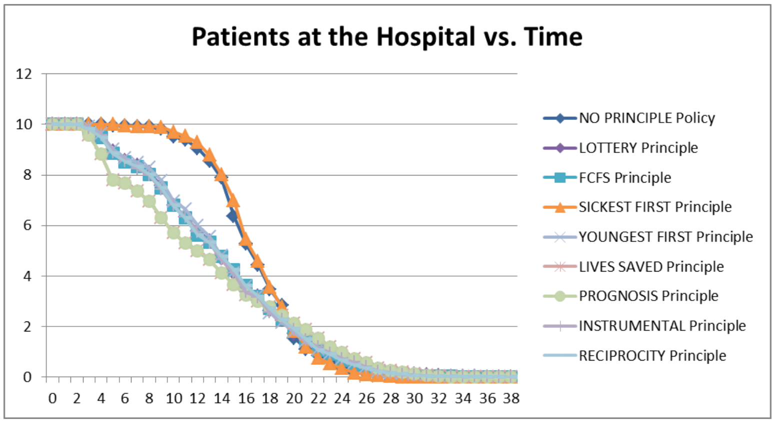

Figure VH-17a – Hospital: Number of Patient at the Hospital over Time (Sensing=1, Buffer=1)

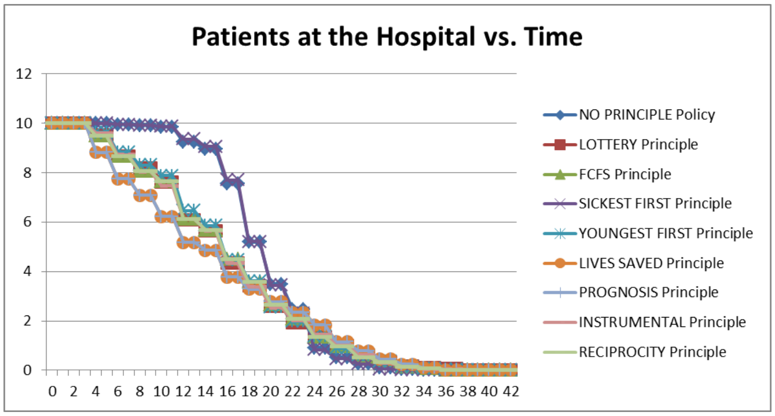

Figure VH-17b – Hospital: Number of Patient at the Hospital over Time (Sensing=2, Buffer=1)

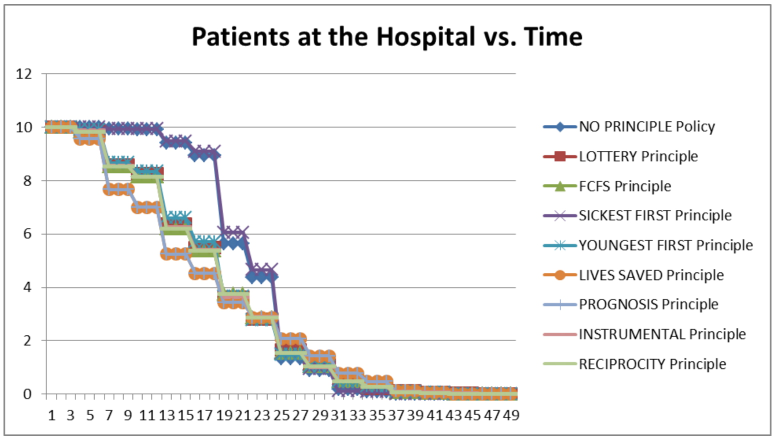

Figure VH-17c – Hospital: Number of Patient at the Hospital over Time (Sensing=3, Buffer=1)

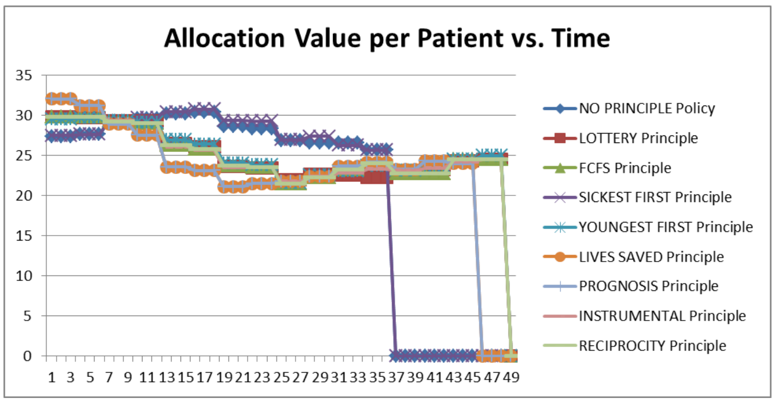

Figure VH-18c – Hospital: Allocation Value per Patient over Time (Sensing=3, Buffer=1)

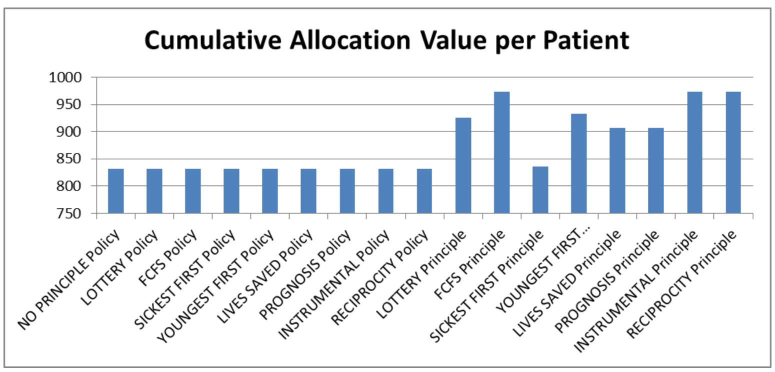

Figure VH-18d – Hospital: Allocation Value per Patient (Sensing=1, Buffer=1)

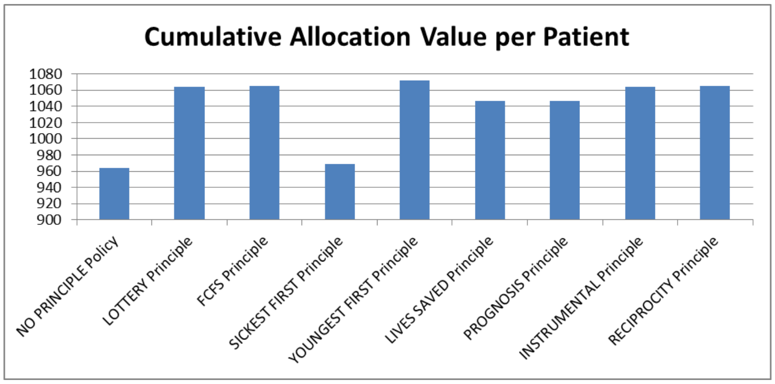

Figure VH-18e – Hospital: Allocation Value per Patient (Sensing=2, Buffer=1)

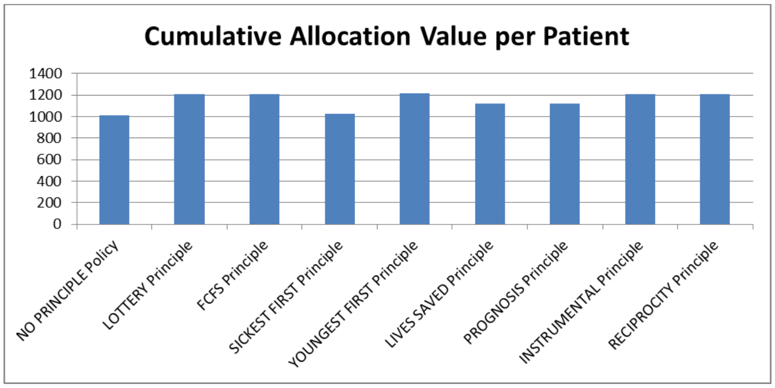

Figure VH-18f – Hospital: Allocation Value per Patient (Sensing=3, Buffer=1)