ABSTRACT

PRADO LOZKO, JORGE ESTEBAN. Tradable Reservations In Multi-Agent, Resource-Constrained Environments. (Under the direction of Peter Wurman)

People waste a lot of time waiting in queues. Even when they go to places where they can make reservations, the reservations alone don't provide much flexibility. The purpose of this research is to show that the use of mobile devices, which allow us to improve planning, trade reservations and receive total information about queues, improves the social welfare. Several experiments were performed using a custom made simulator. The results of the experiments demonstrate that a simple reservation mechanism can even reduce the social welfare under certain conditions, but tradable reservations and clairvoyance each improve it. While in many situations queues are unavoidable, better information and more flexibility in reservation handling can improve the overall quality of societies use of these resources.

Tradable Reservations In Multi-Agent, Resource-Constrained Environments

by

JORGE E. PRADO LOZKO

A thesis submitted to the Graduate Faculty of
North Carolina State University
In partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh

2002

APPROVED BY:

Dr. Michael Young

Dr. Robert St. Amant

Dr. Peter Wurman

Chair of Advisory Committee

Dedication

This thesis is dedicated to my wife, Abelinhe, and to my parents, Elsa and Amadeo, for their unconditional love and support.

Biography

Jorge Prado is originally from Lima, Peru. He pursued his B.S. degree in Informatics Engineering in the Pontifical Catholic University of Peru. He worked as a software developer for three years before he decided to pursue his M.S. degree in the North Carolina State University. He worked part-time as a teaching and research assistant for Dr. Peter Wurman.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Motivation

Every day we get involved in activities in which we have to wait in line to receive some service, or to use some scarce resource. For example, if we go to an amusement park, we may have to wait in line to enter every attraction. We may have to wait at the restaurant to be seated. If we don't know in advance how many people are waiting there, we may get an unpleasant surprise when we arrive. The same scenario is likely to be repeated at many places, including the gymnasium, museums, shopping malls, or any other place where scarce resources are shared.

The problem of waiting in line is solved in part by using reservations. A person can call a restaurant to make a reservation in advance, so he won't have to wait for a table when he arrives. However, reservations are not very flexible. He will have to call the restaurant if a change in the reservation time is required, but if there are no reservations available for the new time, he will have to make a new plan, and maybe choose another restaurant.

Coordinating more than a few reservations may become a non-trivial problem without the help of some decision tool, and even then it may take a lot of our time just planning and getting reservations.

In our research, we propose the use of software agents to help users make better decisions and get the reservations that best match their preferences. A mobile device may be used to communicate user's preferences to the agent, and agent's suggestions to the user. In this thesis, I analyze the system performance, measured in terms of social welfare, as I adjust the following characteristics:

1) **Information Quality**. - Imagine a person who arrives at a restaurant and finds it crowded. If she had known in advance the number of people waiting at the restaurant, she may have chosen another time, or another restaurant. A *clairvoyant* person would know the expected wait time of every restaurant before deciding where to go.
2) **Tradable Reservations**. - Imagine a person who bought the last ticket for the evening show at the theatre. His plan changes suddenly and now he has to catch a plane this afternoon, so he has to get rid of the theatre reservations, and get an airline ticket. He may not be able to buy a ticket from the airline, but there is another traveler who would be willing to cancel his trip if she could get theatre tickets. Markets for the reservations would enable these two people to adjust their plans in a satisfactory manner.
3) **Planning**. - Consider a person in an amusement park who really wants to ride a roller coaster. Acquiring a reservation for the roller coaster right now is impossible, and the queue is really long. If he plans ahead, he may find that the reservations for the roller coaster in a few hours are easy to get, and he can construct a plan to attend other attractions around the roller coaster reservation.

The purpose of this thesis is to investigate by simulation how these three enhancements affect the social welfare. For this purpose, I developed a program that simulates a group of agents in an environment populated by attractions with limited capacity. The model used for the simulations is described in chapter 2. The proposed solution is described in chapters 3 and 4. The simulator is described in chapter 5. The results of the simulations are presented in chapter 6. Related work is presented in chapter 7. The last chapter contains conclusions and future work.

Chapter 2 Model

This chapter describes the model used in the simulations. The goal of the model is to describe an environment that captures key aspects of the real world, and allows us to analyze the behavior of agents inside it.

I consider a discrete time, finite horizon model of the environment. This would be consistent, for instance, with a single day visit to an amusement park, which I use as a prototypical example. A discrete time interval, t, is an element of the ordered set $\{0,\dots,T\}$. The rest of the model consists of two parts: the environment (the simulated world) and the agents (the citizens of our world).

2.1 The Environment

The Environment is the place where activities (theme park rides, museum exhibitions, restaurants, etc.) are located. I model the environment as a connected graph. The nodes of the graph represent the activities (rides, exhibitions, restaurants, etc.). The edges of the graph represent the walkways connecting the activities. The set of all nodes is denoted N and the set of all edges L. Individual nodes and edges are designated n and l, respectively. Agents represent the users of the environment. An individual agent is denoted α, and the set of all agents A.

2.1.1 Node Attributes

Nodes have attributes that govern how agents interact with them. The *maximum capacity* of node n, denoted c_n, is the maximum number of agents that can use activity n at the same time. Naturally, $c_n > 0$. The *duration* is the amount of time an agent spends inside node n, and is represented by $s_n > 0$. The *admittance frequency* is the amount of time between the admittance of one group of agents to the activity and the admittance of the next group of agents. Admittance is periodic starting at $t = 0$. The admittance frequency of node n is denoted f_n, where $f_n > 0$.

These attributes allow us to simulate a wide variety of activities. For instance, roller coasters are modeled as having small capacity, short duration, and frequent admittance. Theatre shows have long duration, relatively large capacity, and infrequent admittance. A sit-down restaurant has moderately frequent admittance and long duration, while a cafeteria may admit one person every time step.

Many of the agents' decisions will require the agent to determine the *next admittance time*–the number of time steps from time t until node n admits another group of agents. Next admittance time is denoted $e_{n,t}$, and can be computed by:

$$e_{n,t} = \begin{cases} 0 & \text{, if } (t \bmod f_n) = 0 \\ f_n - (t \bmod f_n) & \text{, otherwise} \end{cases}$$

Each activity has an ordered set of agents waiting in line to enter the attraction, called the *queue*. The queue at time t, is denoted by $Q_{n,t}$. The agents are represented by α_i, and the position (node location) of agent α_i at time t is $p_{\alpha_i,t}$.

$$Q_{n,t} = \bigcup_i \{ \alpha_i \}, \text{ for all } i, \text{ where } \alpha_i \in A, \; p_{\alpha_i,t} = n, \text{ agent } \alpha_i \text{ is in queue}$$

The *queue length,* $q_{n,t}$, is the number of agents waiting in line to enter node n at time t, and is equal to the cardinality of $Q_{n,t}$. The function $q_{n,t}(\alpha)$ represents the number of people in the line ahead of α.

An agent can compute the amount of time it expects to wait in line for node n with queue length $q_{n,t}$. The expected wait time, denoted $w_{n,t}$, is:

$$w_{n,t} = (\text{floor}[\, q_{n,t} / c_n \,] * f_n) + e_{n,t}$$

If the agent is already in line, the expected time until the agent enters the activity is computed using the same equation, replacing $q_{n,t}$ with $q_{n,t}(\alpha)$.

The queues are FIFO, that is, the nodes receive agents in a first come first served basis (unless the agent has a reservation, described later). If the number of agents who want to enter a node is higher than the maximum capacity of the node ($q_n > c_n$), the first c_n agents enter the node, and the remaining agents ($q_n - c_n$) wait in line. Agents can abandon the line at any time.

The edges of the graph represent walkways (or other means of transportation), that I call links. Each link connects two nodes, and has an associated traversal time, called $d_{m,n}$, where m is the origin node, n is the destination node, and $d_{m,n} > 0$. If nodes m and n are not directly connected, then $d_{m,n}$ is undefined.[1] Note that $d_{m,n} = d_{n,m}$.

The graph generation process used in the simulation ensures that the graph will always be connected, that is, there is always a path from one node to another, using one or more links. Sometimes, the direct link from m to n is not the shortest path (i.e., it may be the scenic route). Let $D_{m,n}$ be the shortest path from m to n, using one or more links. $D_{m,n}$ may be less than $d_{m,n}$. For example, in Figure 2.2, assuming $d_{4,6}$=8 $d_{4,5}$=3, $d_{5,6}$=3 and $d_{4,1}$=10, then $D_{4,6} = 6 < d_{4,6}$.

2.2 The Agents

The agents are simulated people who use the facilities in the environment. An individual agent is denoted α and the set of all agents A.

2.2.1 Agent Attributes

Each agent has attributes that describe its history, its current state, its utility for various actions, and its plan.

---

[1] Links could be modeled as activities with infinite capacity, admittance every time step, and duration $d_{m,n}$.

Agent α's *position* is the node or link at which it is currently located. The agent's position at time t is denoted $p_{\alpha,t}$, where $\alpha \in A$, and $p_{\alpha,t} \in N \cup L$. The agent's *time to finish*, $z_{\alpha,t}$, is the amount of time until agent α completes its current action.

$$z_{\alpha,t} = \begin{cases} D_{n,m} & \text{, if agent } \alpha \text{ enters link } n\text{-}m \\ s_n & \text{, if agent } \alpha \text{ enters activity } n \\ z_{\alpha,t-1} - 1 & \text{, if agent } \alpha \text{ is occupied, and } z_{\alpha,t-1} > 0 \\ 1 & \text{, if agent } \alpha \text{ decides to do nothing for 1 time step} \\ 0 & \text{, otherwise (including standing in line)} \end{cases}$$

The agent keeps track of its *history* in the form of the number of visits it has made to each activity. An agent is considered to have visited a node only if it actually partakes in the activity. The number of times α has entered the activity at node n is denoted $h_{\alpha,n,t}$.

$$h_{\alpha,n,t} = \begin{cases} 0 & \text{, if } t = 0 \\ h_{\alpha,n,t-1} + 1 & \text{, if agent } \alpha \text{ enters the activity at node } n \text{ at time } t \\ h_{\alpha,n,t-1} & \text{, otherwise} \end{cases}$$

The agent derives value by participating in the activities of the environment. The amount of value is defined by the agent's *utility function*. In general, the utility that an agent gets from node n is a function of the number of times that the agent has already visited n. For simplicity, I consider only three types of utility functions, each representing a simplified, but common, scenario in these environments.

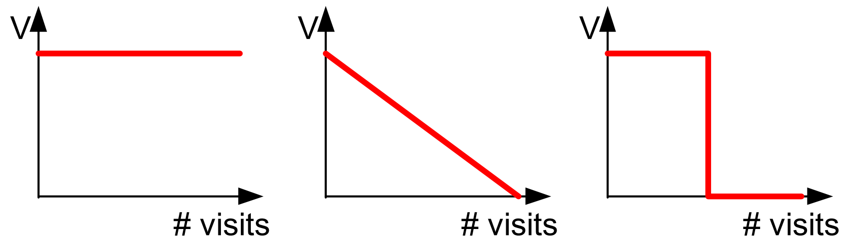

Figure 2.1 - Utility Functions

Figure 2.1 shows the three types of utility functions: (a) utility that is constant, (b) utility that decreases linearly with the number of visits, and (c) utility that is constant to some threshold number of visits, and zero thereafter.

Let $v_{\alpha,n}(h_{\alpha,n,t})$ represent agent α's value for entering n as a function of the number of previous visits. I assume that agents are risk averse, and have quasilinear utility; they are not budget constrained, but they do have some other use for money outside the environment. In practice, I expect that it will be more acceptable to be granted a budget in a currency local to the environment.

Agents are utility maximizers and will follow the plan that gives them the highest utility in their planning horizon.

The accumulated utility earned by agent α from time t_0 until time t is represented by $U_{\alpha,t}$.

$$U_{\alpha,t} = \sum_{n} \sum_{i=1}^{h_{\alpha,n,t}} v_{\alpha,n}(i)$$

I assume that agents don't share information between them. One agent doesn't know the value or the plans of other agents, and it does not try to predict what the other agents will do next.

2.2.2 Agent State

The agent's current state is the agent's position and the agent's current action. The possible actions are:

Begin Walk: An agent can move from one place to another when it is not inside an attraction. Once it decides to go to another place, it cannot change actions until it reaches another node. For example, if an agent is going from A to D, using the links A-B, B-C and C-D on its way, it can only change its destination when reaching B, C or D. All agents walk at the same pace; there are no faster or slower agents. Agents always use the shortest paths (similar to having a map).

Enter Queue: When an agent arrives at a node and wants to enter, it has to go to the end of the line and wait its turn. When two or more agents arrive to a queue at the same time, the tie is broken randomly.

Enter Activity: When the agent is within c_n of the front of the queue and the node is admitting users, the agent enters the activity. The amount of time the agent will be inside node n is s_n, and the agent is removed from the queue. After entering a node, agents cannot leave it before the activity is over.

Wait: An agent may prefer to wait in some place, doing nothing.

2.2.3 Agent Perception

The *perception* of the agent determines how far the agent can see. Specifically, it represents the list of places where the agent can see the queue length, and can compute the expected wait time. Two different perception options are analyzed:

A *myopic* agent can see only its current location, and the neighbors that surround it (nodes directly connected to the node where the agent is located). If the agent is located in node *n*, then it can see the queue at node *n* and the queues at all nodes *m* where $d_{n,m}$ exists.

Clairvoyant agents can see the queue length of all the activities.

In Figure 2.2, an agent in node 1 with myopic perception can see the queue lengths of its current position (node 1) and the immediate neighbors (nodes: 2, 3, 4 and 7). With clairvoyance, it would see all the nodes.

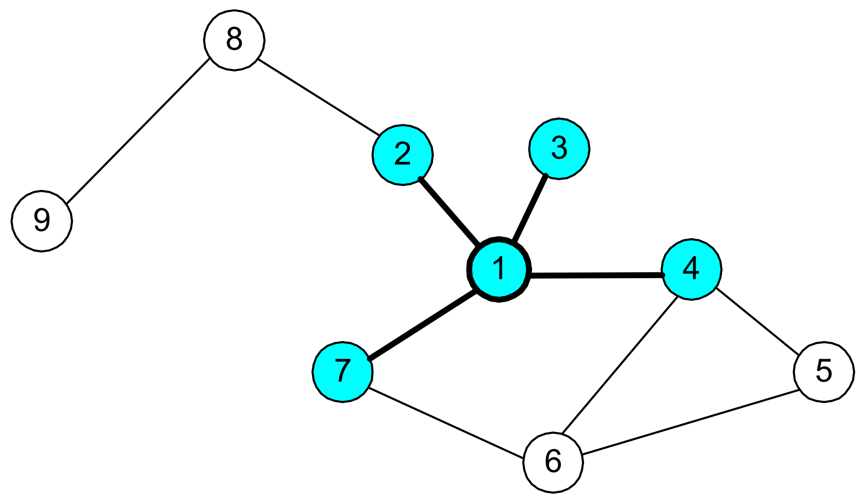

Figure 2.2 - Myopic and Clairvoyant Perception

Chapter 3 Enhanced Environments

I enhance the environments modeled in the previous chapter by adding tradable reservations and clairvoyance to improve planning. The reservations can be exchanged in markets, and I assume that the agents (on mobile devices) are endowed with communication technology that enables them to participate in the market while being carried around the environment.

3.1 Tradable Reservations

Reservations guarantee (in principal) that a certain service will be provided to you at a certain time. A reservation can only be used at a specific time, and has no value after that. A *reservation* for node n at time t is denoted $r_{n,t}$. If an agent holds $r_{n,t}$, it will be admitted to node n at time t without waiting in line. The number of reservations distributed in node n for each admittance time is represented by r_n, $0 \leq r_n \leq c_n$.

If an agent doesn't hold a reservation, it must account for the effect that other agent's reservations will have on its wait time. The expected time spent in line for an agent that doesn't have a reservation is

$$w_{n,t} = (\text{floor}\,[\, q_{n,t} / (c_n - r_n)\,] * f_n\,) + e_{n,t}$$

If the agent is already in line, $w_{n,t}$ is computed using the same equation, replacing $q_{n,t}$ with $q_{n,t}(\alpha\,)$. If the agent has a reservation for the next admittance time, its wait time $w_{n,t}$ will be $e_{n,t}$.

I refer to the reservations that an agent currently holds as its *endowment*, denoted $E_{\alpha,t}$. During the trading stage of iteration t, the agent is free to sell part or all of its endowment. Reservations that the agent purchases during the trading stage are either used immediately or added to the agent's endowment for the next iteration.

Reservations can be traded in an electronic marketplace. The market has one auction for each possible node and time where reservations are possible. As a practical matter, the number of reservable times may be regulated to keep the number of markets down. For instance, the markets may be opened for only the next two hours worth of activities, or only for the entry to activities on the quarter hour mark. The market price of a reservation to node n at time t is denoted $\pi_{n,t}$.

3.2 Planning

At each decision point, the agent evaluates its possible actions and selects a plan, denoted *P,* which generates the greatest utility. I assume the agent's planner generates only feasible plans.

Suppose agent α is at node m and plans to go to node n. The value of going to node n at time t is $v_{\alpha,n}(h_{\alpha,n,t})$. The total amount of time that α will spend to complete the activity of node n is the time to finish its current activity $z_{\alpha,t}$ at node m, plus the travel time $D_{m,n}$, plus the time in line $w_{n,t}$, plus the duration s_n of n. Thus, the utility of the plan to go to node n is the value of the plan per time step:

$$u_{\alpha,t} = v_{\alpha,n}(h_{\alpha,n,t}) / (z_{\alpha,t} + D_{m,n} + w_{n,t} + s_n)$$

When using tradable reservations, the price of the reservations $\pi_{n,t}$ impacts the utility as described below.

Suppose agent α is at node *m* and plans to go to node *n* without a reservation. In addition to the value of the plan per time step, the utility that α receives from a plan will also include the market value of the unused portion of the agent's endowment (i.e., the sum of bid prices of everything that it owns). Thus, the utility of the plan to go to node *n* is

$$u_{\alpha,t} = (v_{\alpha,n}(h_{\alpha,n,t}) / (z_{\alpha,t} + D_{m,n} + w_{n,t} + s_n)) + \sum_{\substack{n',t' \\ r_{n',t'} \in E_{\alpha,t}}} \pi_{n',t'}$$

If the agent decides to use a reservation that it owns, the time in the queue will be the time to next admittance $e_{n,t}$, and the income from the agent's endowment will not include the reservation that it intends to use. The utility will be

$$u_{\alpha,t} = (v_{\alpha,n}(h_{\alpha,n,t}) / (z_{\alpha,t} + D_{m,n} + e_{n,t} + s_n)) + \sum_{\substack{n',t' \\ r_{n',t'} \in E_{\alpha,t}}} \pi_{n',t'} - \pi_{n,t}$$

If the agent needs to buy the reservation, *r*, that completes the plan, the utility expression will have the same form as the above with the exception that *r* is not an element of $E_{\alpha,t}$.

3.2.1 Complex Plans

When planning farther ahead, it is possible that a plan will require more than one reservation. To compute the utility of a complex plan *P*, the agent computes the average value of all the activities considered in the plan horizon. Denote the average value of a complex plan *V*. The utility will be

$$u_{\alpha,t} = V + \sum_{\substack{n',t' \\ r_{n',t'} \in E_{\alpha,t}}} \pi_{n',t'} - \sum_{\substack{n'',t'' \\ r_{n'',t''} \text{ used in plan } P_{\alpha,t}}} \pi_{n'',t''}$$

3.2.2 Breaking Ties

When two plans provide the same utility to an agent, the agent selects one according to the following rules:

- A plan that uses reservations has priority over plans that consider going to wait in line (because of the uncertainty of the future queue length).
- A plan that uses a reservation that the agent owns has priority over a plan that considers buying reservations.
- Doing nothing has the lowest priority.

Chapter 4 The Auction Mechanism

An auction is a mechanism used to reallocate goods between buyers and sellers. There are four classic types of auctions: English (open, ascending auction), Dutch (descending auction), first-price sealed bid auction and Vickrey auction [8].

General Equilibrium Theory provides a set of sufficient conditions that ensure the existence of equilibrium prices and the Pareto optimality of the supported allocation [7]. Unfortunately, the exchange economy defined by the agents in our model does not satisfy the conditions of the First Welfare Theorem. In particular, the goods are discrete and therefore violate the condition that preferences be convex. In addition, as the planning horizon increases, agents begin to construct plans that include complementary reservations, which violates the gross substitutes condition. The presence of these two violations will sometimes prevent the markets from converging. For the purposes of this study, which is essentially to show the impact of trading on planning and overall social welfare, I have manipulated the auction to improve convergence at the expense of optimality.

4.1 Auction Rules

The reservations are traded in k-double auctions where $k = 1/2$. I assume that agents bid truthfully and state their actual willingness to pay for (or minimum willingness to sell) a particular reservation. Moreover, I assume the agents act competitively and take the prices announced by the auctioneer at face value.

The price quote generated by the auction is computed according to the standard M^{th} and $(M+1)^{st}$ price rules. The buy and sell bids are sorted, and the M^{th} and $(M+1)^{st}$ highest bids are identified, where M is the number of sell offers. These prices delineate the range of prices that will balance supply and demand. The M^{th} price corresponds to the ask quote, π_{ask}, and the $(M+1)^{st}$ price is the bid quote, π_{bid} [12].

After all the bids have been received, new quotes are computed and communicated to the agents. Agents can then update their bids in response to the new prices. The auction will reach equilibrium when no agent wants to change its bids, given the current prices of the reservations.

Leon Walras introduced the idea that the process of receiving bids, announcing price quotes and allowing the users to adjust their bids - a process called Walrasian tatonnement - would lead to equilibrium, the condition when no user wants to change his bid. The algorithm is shown in [3].

If the market fails to reach equilibrium, I introduce error in the utility computed by the agents by manipulating the price quotes. When a convergence failure occurs, defined by the situation where there is no price vector that makes all the agents keep their bids unchanged, the market will modify the prices announced by adding (subtracting) ε to the ask (bid) quote. This has the effect of making reservations seem more expensive to buyers, and less valuable to sellers. When considering buying *r*, agents will be told the ask price of *r* is not π_{ask} but $\pi_{ask} + \varepsilon$, thus decreasing their expected utility of buying it. When considering selling *r*, agents will use the bid quote $\pi_{bid} - \varepsilon$, decreasing their expected utility of selling it. If the market still fails to converge, ε is increased. Eventually the announced prices will reach values where no agent wishes to place a new bid, which I call quiescence, but in so doing the market sacrifices social efficiency.

After reaching quiescence, the markets clear. The exchange price is determined by using $k = 1/2$, that is, the middle point between the bid and ask quotes. All the sellers with bids below the exchange price will sell, and all the buyers with bids above the exchange price will buy. The sellers will transfer the reservation to the buyers, and buyers will reciprocate with money in the amount equal to the trading price. The detail of who exchanges the goods with whom is not important, because all the goods in a specific market are identical.

4.2 Bidding

Agents interact with the market by placing bids to buy and sell reservations. The first step in determining bids is to compute the utility of each plan (see section 3.2). I will also assume that the agents are truthful and bid their exact willingness to pay (sell). In addition to being truthful, the agent is pessimistically using the bid and ask quotes rather than computing the true $k = 1/2$ transaction prices.

Each agent will find the plan that gives him the highest utility. I will denote the highest utility $u_{\alpha,t}$* and the next highest utility $u_{\alpha,t}$'. Let V^* be the average value of the plan and π^* the price of the reservation used in the highest utility plan, as calculated before. Similarly, V' and π' represent the valuation and the price of the reservation used in the second highest utility plan.

The value that α has for a reservation, *r*, is the amount that would make the agent indifferent between the plan with *r* and the best alternative. To compute a bid, $b_{\alpha,n,t}$, for a reservation, r^*, that is part of the best plan, the agent compares the plan to the second best plan. The agent is indifferent when $u_{\alpha,t}^* = u_{\alpha,t}'$. Therefore,

$$V^* - \pi^* = V' - \pi'$$

$$\pi^* = V^* - (V' - \pi')$$

Thus, the truthful agent will bid $b_{\alpha,n,t} = V^* - (V' - \pi')$. If $r^* \in E_{\alpha,t}$, α is willing to sell r^* as long as it receives at least $b_{\alpha,n,t}$. If r^* is not in α's endowment, α is willing to buy r^* for up to $b_{\alpha,n,t}$.

Bids for reservations that are part of the other plans are assessed in the same manner. Let $u_{\alpha,t}$ be the utility of a arbitrary other plan. Again, the agent finds the indifference point where $u_{\alpha,t}^* = u_{\alpha,t}$. It follows that

$$V^* - \pi^* = V - \pi$$

$$\pi = V - (V^* - \pi^*)$$

And the truthful bid will be $b_{\alpha,n,t} = V - (V^* - \pi^*)$.

Table 4.1 shows an example of the bids computed by an agent, assuming the agent is in node 1 at time 0 and there are only 3 nodes in the graph, all connected. The price of the reservation for ride 1 at time 1 (that he owns) is π_1^{bid}, for ride 2 at time 2 is π_2^{ask} and for ride 3 at time 1 is π_3^{ask}. Also $z_{\alpha,t} = 0$, and $5 - \pi_1 > 3.75 - \pi_2 > 2.14$, that is: $\pi_2 < 1.61$ and $\pi_1 < 1.25 + \pi_2$. These conditions ensure that the plan "Ride 1 with reserv." has the highest utility.

Table 4.1 - Example of Calculation of Bids

Plan	$D_{m,n}$	$w_{n,t}$	s_n	$V_{n,t}$	V	$V- \pi$	Offer	$b_{\alpha,n,t}$
Ride 1 with reserv.	0	1	1	10	10/2 = 5	$5 - \pi_1$	Sell $r_{1,1}$	$5 - 3.75 + \pi_2$
Ride 2 with reserv.	2	0	2	15	15/4 = 3.75	$3.75 - \pi_2$	Buy $r_{2,2}$	$3.75 - 5 + \pi_1$
Ride 2	2	3	2	15	15/7 = 2.14	2.14		
Ride 3 with reserv	1	0	3	8	8/4 = 2	$2 - \pi_3$	Buy $r_{3,1}$	$2 - 5 + \pi_1$
Ride 1	0	5	1	10	10/6 = 1.66	1.66		
Ride 3	1	1	3	8	8/5 = 1.6	1.6		

4.2.1 Bidding in Multiple Markets

The process is more difficult when plans involve more than one activity because of the difficulty of ascribing the value of a plan to the component reservations. One solution is to permit combinatorial bidding [5]. However, due to the size of real problem instances, that may prove intractable.

I ascribe value to component reservations with the following heuristic. Let the superscript # denote the highest utility plan that uses *r*. Let $\pi_{others}^{\#}$ denote the sum of the market prices of all other reservations required in plan #. Using the same nomenclature presented before, the amount to bid for each reservation in the plan will be:

$$u_{\alpha,t}^{\#} = u_{\alpha,t}'$$

$$V^{\#} - \pi_{n,t}^{\#} - \pi_{others}^{\#} = V' - \pi'$$

$$\pi_{n,t}^{\#} = V^{\#} - \pi_{others}^{\#} - (V' - \pi')$$

Thus, the truthful bid will be $b_{\alpha,n,t} = V^{\#} - (V' - \pi') - \sum_{\substack{n',t' \\ r_{n',t'} \text{ used in plan } P_{\alpha,t} \\ (n,t)\neq(n',t')}} \pi_{n',t'}^{\#}$.

The same applies when bidding in plans that don't give us the maximum utility:

$$b_{\alpha,n,t} = V - \pi_{others} - (V^{\#} - \pi^{\#})$$

$$b_{\alpha,n,t} = V - (V^{\#} - \pi^{\#}) - \sum_{\substack{n',t' \\ R_{n',t'} \text{ used in plan } P_{\alpha,t} \\ (n,t)\neq(n',t')}} \pi_{n',t'}$$

All the agents will place their bids: buy bids for reservations they don't have, and sell bids for reservations that they have. If an agent holds a reservation that it cannot use in any of its plans (because, say, it can't get to the node by the reservation time), it will offer to sell the reservation for 0.

4.3 Finding Equilibrium

The next step will be to compute the prices that balance supply and demand. To do that, I apply the k-double auction rules.

After all the bids have been received, new quotes are computed and communicated to the agents. Agents can change their bids considering the new prices.

The auction will reach equilibrium when no agent wants to change his bids, given the current prices of the reservations.

Chapter 5 The Simulator

The simulator is a program written in LISP, with 2 main functions: It creates scenarios and agents with random characteristics, and it executes the simulation, allowing the agents to interact with each other and with the environment, according to rules explained later.

5.1 Creating a Random Problem Instance

The program is extensively parameterized. To create a random problem instance, the program receives the following information:

- Number of nodes
- Number of agents
- Number of time steps

Then the program populates the nodes, links and agents by selecting values randomly chosen from a uniform distribution, according to parameters, most of them ranges of values to use. The parameters used are:

- The range of values for max. capacity c_n, time inside s_n, admittance frequency f_n, distance $d_{m,n}$, the initial value of each node for each agent $V_{\alpha,n}(0)$ and the type of utility function of each agent.
- The percentage of all the possible links that should be added to the model, in addition to the minimum number of links (number of nodes – 1), which are added automatically to ensure that the graph is connected.
- The percentage of the total capacity of each node that will be reservable r_n

The agents are located at a randomly chosen node. Once the agents and nodes are created, the program distributes reservations to agents chosen randomly. The behavior is different if the non-random distribution of reservations was specified (see section 5.4).

After populating all the data structures, lists containing all the agents in random order are created for each time step. The lists will be used as the execution order and will determine how the ties are broken when two agents arrive to a node at the same time.

5.2 Parameters of the Simulation

The parameters that can be specified to change the behavior of the simulator are:

- Agent perception (myopic – clairvoyant)
- Planning horizon (number of time steps to plan ahead)
- Enable / disable distribution of reservations
- Enable / disable non-random distribution of reservations
- Enable / disable trade of reservations

5.3 Running the Simulation

The main loop of the simulator is shown in figure 5.1. First, the simulation is initialized. Then, each agent generates his plan and, if trading is enabled, negotiates in the market bidding truthfully (see chapter 4). The plan that generates maximum utility to the agent is executed and the loop is repeated.

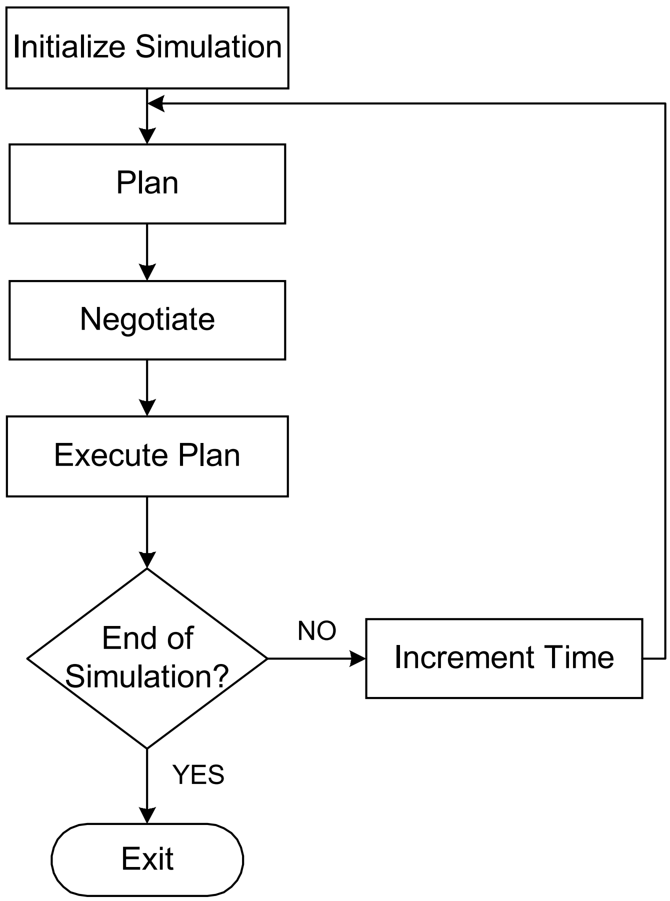

Figure 5.1 - Main Loop of the Simulation

5.3.1 Initialize Simulation

When the simulation starts, the problem instance is loaded in memory or a new instance is created, according to the parameters specified. The minimum distance $D_{m,n}$ and minimum paths between all pairs of nodes is computed using the Floyd-Warshall algorithm [6]. The lists of activities that are visible from each node are computed according to the perception parameter specified, and stored in memory. Each agent is randomly positioned next to an activity.

5.3.2 Generate Plans

For each agent, the possible plans are generated, according to the planning horizon specified. The plans generated by each agent can only contain nodes in their perception scope (see section 2.2.3), or plans that consider using reservations that the agent holds or can buy. The plans also consider doing nothing in any time step. That feature allows agents to consider not only the next admittance time of any node, but future ones where reservations may be cheaper. The agent computes the value of each plan while taking into consideration the queues and availability of reservations.

5.3.3 Negotiation

The agents bid truthfully, according to the extra utility (surplus) they would earn if winning the auction, as described in section 3.3.

After receiving the bids of all the agents, the auction mechanism computes the new prices. The cycle is repeated until quiescence is achieved.

5.3.4 Execution

Each agent chooses the plan that will give him the highest utility, according to his current endowment. Figure 5.2 shows the main steps in the execution of the agent's plans. If the agent is inside an attraction or walking, nothing is done, because those actions cannot be interrupted. If the agent is not at the node that he wants to visit, the agent walks to the next node in the shortest path to arrive at the destination node.

The simulator executes agents' plans in a random sequential order. If many agents plan to go to the line of node n, the agent chosen first will "arrive" to the node first. This randomization ensures that no agent is always preferred to another.

When the agent arrives to the desired node, he goes to the line, or goes directly to the front if he owns a reservation for that time period.

After all the agents' plans are executed, the queues of all the nodes are processed. That process consists of taking the first c_n agents from the front of the queue and putting them inside the attraction. The group of agents that enter the attraction includes the agents that have a reservation for that time, which are at the front of the line. If some reservable capacity is not used, more agents without reservations are allowed to enter until all the available capacity is used.

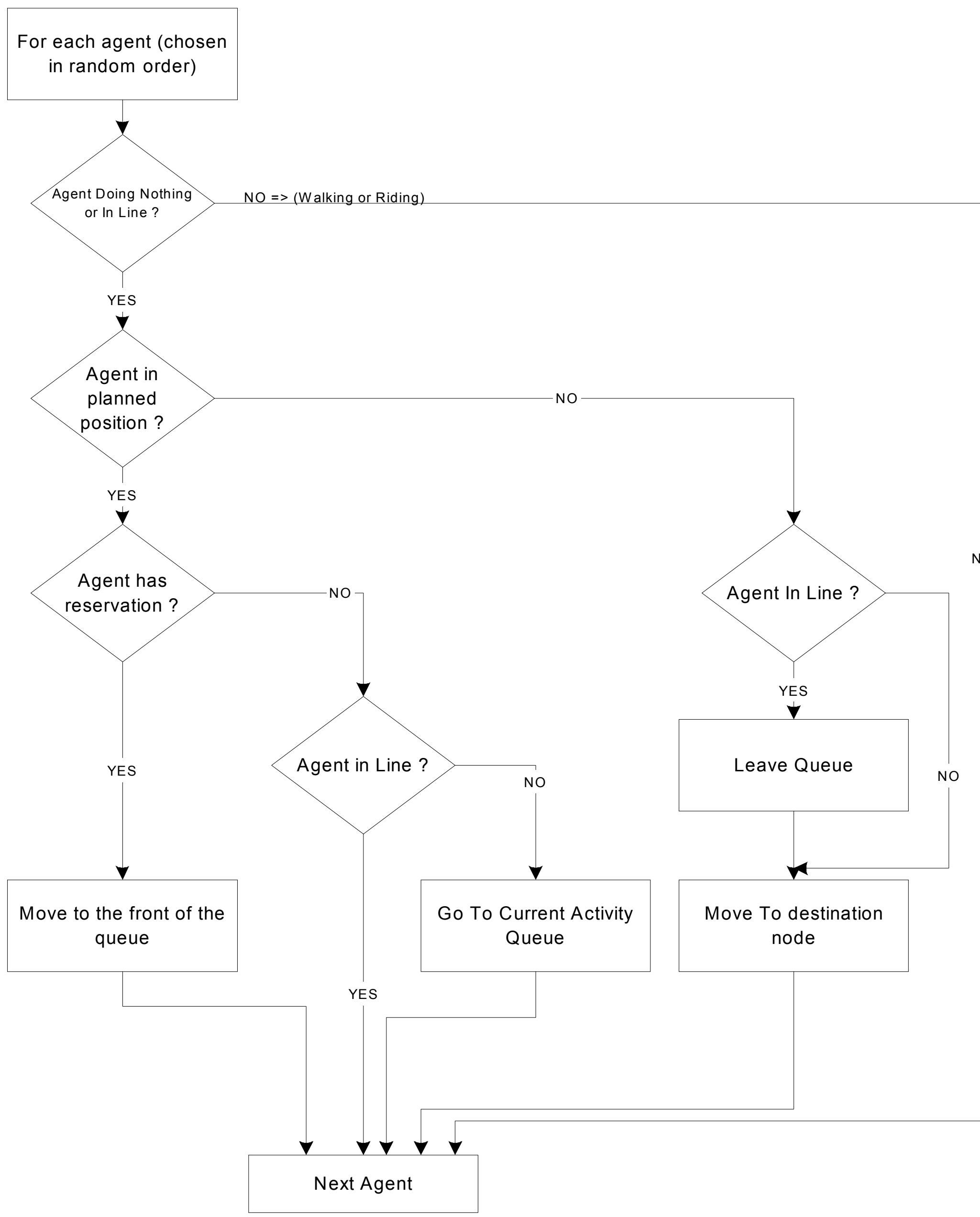

Figure 5.2 - Execution Diagram

5.3.5 Increment Time

This process marks the end of the simulated time step. The time is incremented and if the number of time steps to simulate has not been reached, the next time step is simulated.

5.4 Test Scenarios

The following variations in the behavior of the simulator were studied:

No Trading, No Reservations. - No reservations are distributed. There is no reservable capacity in any node. No trading is allowed.

No Trading, Random Reservations. - Reservations are distributed randomly among agents at the beginning of the simulation. No trading is allowed. I have observed by running a series of random simulations that the average usage of reservations is 9%, so when computing the wait time agents assume that only 9% of the reservations will be used.

No Trading, Non-Random Reservations. - Reservations are given for free to agents that request them. No trading is allowed. The reservations usage is also assumed to be 9%. The non-random distribution of reservations consists of distributing reservations to the agents that ask for them, during the execution of the simulation. An agent asks for a reservation when the reservation is part of his maximum utility plan, and all the reservations required to execute that plan are available. If they are not available, the agent asks for the reservations of the next best utility plan, and so on until he has a plan that he can execute.

Trading, Reservations. - Reservations are distributed for free randomly, but once they are assigned trading is allowed.

Chapter 6 Experiments and Results

6.1 Experiments

Several experiments where performed using the simulator. For each series of experiments, the procedure is described below.

Two problem instances with random data were created. The ranges of values for each of the parameters used were:

- Initial value of agent α entering each node n, $v_{\alpha,n}(0)$ in [0, 10]
- Max. Capacity c_n in [1, 10]
- Time Inside s_n in [1, 2] (s_n=1 in the second problem instance)
- Distance $d_{m,n}$ in [1, 3] ($d_{m,n}$=1 in the second problem instance)
- Admittance Frequency f_n in [1, 3] ([1,2] in the second problem instance)
- Links 15 - 30%
- Linear utility functions $v_{\alpha,n}(x)= v_{\alpha,n}(0) * (1 – bx)$, with b in [0.0, 1.0]

The problem instances have 10 nodes, 100 agents and 100 time steps. The graphs of the environments are presented in figures 6.1 and 6.2. The attributes of the nodes are presented in tables 6.1 and 6.2. The total capacity of the first instance is 36.24 agents per time step. Thus only slightly more than a third of all agents can be active at one time. The links in the first problem instance have $d_{m,n}$=2, except the links 1-4, 3-4 and 5-7 with $d_{m,n}$=1, and the links 3-9, 5-8 and 5-6 with $d_{m,n}$=3. The second problem instance has shorter activities, higher admittance frequency and more links. The total capacity of the second problem instance is 43.00 agents per time step. All the nodes of this instance have $d_{m,n}$=1.

I measured the social welfare of the system under the four test scenarios described in section 5.4. The four types of simulations were run with both myopic and clairvoyant perception and with varying levels of reservable capacity (20%, 40%, 70% and 100%) and planning horizon (1 to 4 time steps).

Table 6.1 - Attributes of the Nodes of the First Problem Instance

Node	1	2	3	4	5	6	7	8	9	10
f_n	3	2	2	1	1	3	1	2	3	1
c_n	1	5	4	10	4	7	4	10	3	5
s_n	1	2	2	2	1	1	2	2	2	1

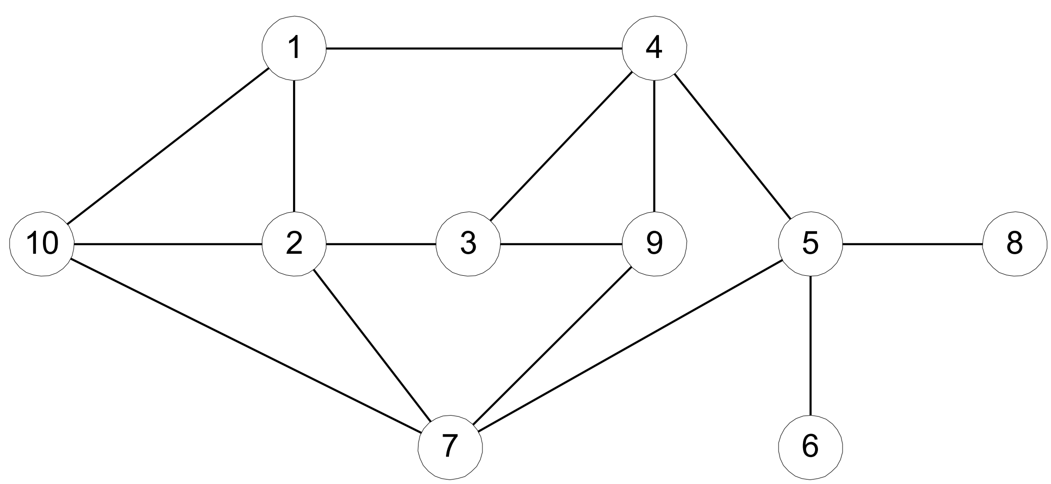

Figure 6.1 - Graph of the First Problem Instance

Table 6.2 - Attributes of the Nodes of the Second Problem Instance

Node	1	2	3	4	5	6	7	8	9	10
f_n	2	1	2	2	1	2	2	1	2	1
c_n	7	4	2	8	4	10	7	8	2	9
s_n	1	1	1	1	1	1	1	1	1	1

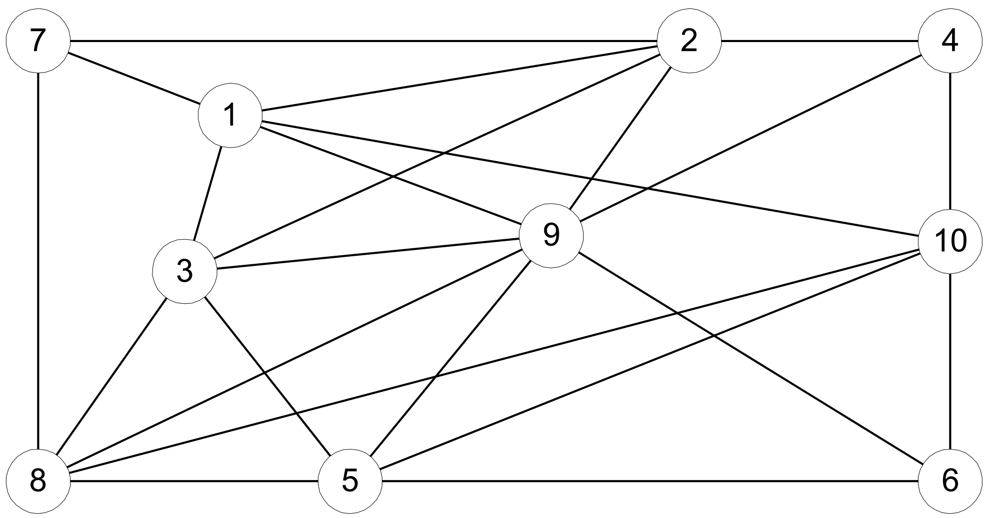

Figure 6.2 - Graph of the Second Problem Instance

6.2 Results

The results of the simulations of the first problem instance are shown in Tables 6.3 and 6.4. For each type of simulation, the utility and percentage of reservations used by the agents is presented. It is interesting to note the increase in reservations usage between the different types of simulations. For example, in the myopic simulation with 20% of reservable capacity (first column), reservation usage increases from 7.02% with random reservations, to 54.75% with tradable reservations. Figures 6.3 to 6.6 compare the social welfare of the 8 types of simulations when 20%, 40%, 70% and 100% of the capacity is reservable (respectively). Each point in the graphs represents the social welfare of the agents in one experiment. Each line represents a different set of parameters used in the simulations. Note that the lines that represent the simulation using no reservations do not change when varying the reservable capacity – in that case the reservable capacity is always assumed zero. Also note that in the simulation with 100% of reservable capacity, the three different reservation allocation mechanisms are independent of the perception – they have the same values with myopic and clairvoyant perception. Figure 6.7 shows a set of Clairvoyant – Trading simulations varying the reservable capacity. It is clear from that graph that, for the problem instance analyzed, more reservable capacity increases the social welfare. Note that convergence problems decrease the utility, and these problems are more frequent when the reservable capacity and planning horizon are increased. This implies that the results are a lower bound on the benefits of trading.

Similarly, Tables 6.5 and 6.6 and Figures 6.8 to 6.12 present the results of using the second problem instance. The difference between myopic and clairvoyant agents is very small in this problem instance. The reason is the larger number of links between nodes. A clairvoyant agent will see in average 4 more nodes than a myopic agent, compared to the 6 extra nodes a clairvoyant agent will see in the first problem instance. Clairvoyance allows agents to consider more activities when planning, but can also make them walk longer to find an unexpected long queue, which will decrease its utility. That negative effect is not clear in the first problem instance because of the large benefit obtained by the increased visibility, but is very clear in the second problem instance. There are so many links between nodes, and the distances are so short that many agents will notice short lines at the same time and will arrive there very fast. The effect is so important in the second problem instance that myopic solutions have higher utility than clairvoyant ones.

Figure 6.13 shows the evolution of the queue length of 6 of the 10 nodes of the first problem instance, when using no reservations with myopic perception and planning 1 step ahead. Each line represents the queue length of one node at the end of each time step, immediately after the agents are admitted to the activities. It is interesting to note how the queues lengths of some nodes oscillate between high and low values. That behavior is caused because those nodes have admittance every other time step. Nodes 1 and 10 have the smallest capacity of the environment. After some time, many agents who still didn’t enter those activities will have a high value for them. That explains why the lines keep getting longer at those nodes.

Figure 6.14 shows similar information when trading is allowed and 40% of the nodes’ capacity is reservable. Agent’s behavior is very similar to the one described before. The queue length of node 1 is longer, being it the smallest capacity node in the environment. Many agents using the activities with higher capacity earlier in the simulation cause this behavior (with clairvoyant perception more agents can use their time better), and then more of them will be willing to wait longer to enter this activity, because their value of the other activities will be lower.

Figure 6.15 analyzes the evolution of prices over time. The graph shows the first problem instance using Clairvoyance, trading and planning 1 time step ahead. The dotted line shows the queue length of the node analyzed. Each continuous line represents the price of a reservation over time. It is clear from the graph that the longer queue length makes the reservations more valuable, and that price goes up as reservation time nears.

Several trends are visible in the graphs. First, the difference between the myopic results and clairvoyant results is quite large in the first problem instance. This result primarily reinforces the importance of making good information available to the agents. Second, with or without clairvoyance, trading improves social welfare. However, adding reservations without trading sometimes decreases the social welfare because a large percentage of them go unused, which adds noise to the estimation of wait times. Finally, increasing the planning horizon does not clearly help or hurt. This inconclusive result is due, in part, to the fact that the horizons are all quite small and the planning algorithm is not sophisticated.

The availability of reservations without trading improves the utility in the first problem instance, but in some cases decreases it in the second problem instance. The usage of those reservations is usually around 10%, and this is because it is sometimes impossible for an agent to use sets of reservations it has been allocated, like two reservations at almost the same time in very distant nodes.

Non-randomly distributed reservations give better results than randomly distributed reservations when the planning horizon is small. When planning ahead, more reservations requested by the agents are not used and wasted, because they change their plans and the unused reservations cannot be reallocated.

There are two sources of “noise” in the agent’s decision process that may help explain the results.

1) Uncertain queue length: When an agent decides to go to a node without a reservation, the queue length will change by the time it arrives. If many agents converge on a node because they heard the line was short, a good portion of them will be disappointed. This situation may arise even when most of the capacity of the nodes is reservable.

2) Convergence problems: Each time the markets fail to converge, a bias is introduced in the quotes to force convergence (see section 4.1). This bias can produce a sub-optimal allocation. I expect the convergence problems to be more frequent when the planning horizon is increased.

Even with those sources of noise, it is clear that using tradable reservations improves the social welfare.

Table 6.3 - Results of First Problem Instance - Myopic Simulation

Planning Horizon	Type of Simulation	20% Reservations		40% Reservations		70% Reservations		100% Reservations	
		Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used
1	No Reservations	8211.20		8211.20		8211.20		8211.20	
2	No Reservations	8210.75		8210.75		8210.75		8210.75	
3	No Reservations	8264.01		8264.01		8264.01		8264.01	
4	No Reservations	8177.57		8177.57		8177.57		8177.57	
1	Random Reservations - No Trading	8638.29	7.02	9011.74	7.35	9031.54	7.62	5302.05	32.86
2	Random Reservations - No Trading	8605.38	8.06	9073.69	8.23	9334.21	8.26	5476.13	33.25
3	Random Reservations - No Trading	8711.27	7.44	9235.25	9.42	9077.57	9.03	5584.49	33.50
4	Random Reservations - No Trading	8664.45	8.47	9194.59	10.46	9331.87	9.17	5671.34	33.66
1	Non-Random Reservations - No Trading	8897.37	5.99	9103.14	5.91	9201.13	6.08	3369.56	16.72
2	Non-Random Reservations - No Trading	8849.02	7.64	9129.65	5.43	9324.28	5.58	2901.39	14.93
3	Non-Random Reservations - No Trading	8610.40	8.68	9223.06	6.71	9311.58	6.99	3013.01	15.20
4	Non-Random Reservations - No Trading	8454.47	5.09	9149.05	8.23	9167.71	6.26	3156.63	16.67
1	Reservations - Trading	9468.96	54.75	9693.89	52.40	9724.55	44.65	9998.37	64.02
2	Reservations - Trading	9548.10	53.09	9498.28	53.36	9720.65	46.55	9855.90	63.66
3	Reservations - Trading	9606.85	59.08	9578.24	57.75	9690.74	46.47	9916.59	63.36
4	Reservations - Trading	9569.04	60.94	9390.51	54.79	9699.36	45.20	9743.72	64.04

Table 6.4 - Results of First Problem Instance - Clairvoyant Simulation

Planning Horizon	Type of Simulation	Reservations = 20%		Reservations = 40%		Reservations = 70%		Reservations = 100%	
		Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used
1	No Reservations	9575.69		9575.69		9575.69		9575.69	
2	No Reservations	9582.06		9582.06		9582.06		9582.06	
3	No Reservations	9590.71		9590.71		9590.71		9590.71	
4	No Reservations	9541.69		9541.69		9541.69		9541.69	
1	Random Reservations - No Trading	9550.73	3.72	9487.11	6.23	9644.22	6.08	5302.05	32.86
2	Random Reservations - No Trading	9515.72	3.72	9486.24	6.55	9635.27	7.08	5476.13	33.25
3	Random Reservations - No Trading	9517.43	5.58	9528.24	7.19	9595.49	6.03	5584.49	33.50
4	Random Reservations - No Trading	9511.53	4.34	9541.67	7.59	9522.86	6.08	5671.34	33.66
1	Non-Random Reservations - No Trading	9548.64	3.72	9579.67	3.04	9639.75	3.22	3369.56	16.72
2	Non-Random Reservations - No Trading	9525.68	4.34	9576.11	3.75	9534.31	3.49	2901.39	14.93
3	Non-Random Reservations - No Trading	9515.53	4.34	9595.91	4.55	9647.40	4.67	3013.01	15.20
4	Non-Random Reservations - No Trading	9521.07	5.40	9562.44	4.97	9506.41	4.22	3156.63	16.67
1	Reservations - Trading	9566.41	42.36	9695.17	46.73	9721.27	43.60	9998.37	64.02
2	Reservations - Trading	9589.51	38.65	9655.12	49.05	9779.99	45.37	9855.90	63.66
3	Reservations - Trading	9516.74	43.19	9662.27	46.89	9764.39	48.77	9916.59	63.36
4	Reservations - Trading	9628.59	45.26	9700.88	47.20	9766.74	47.91	9743.72	64.04

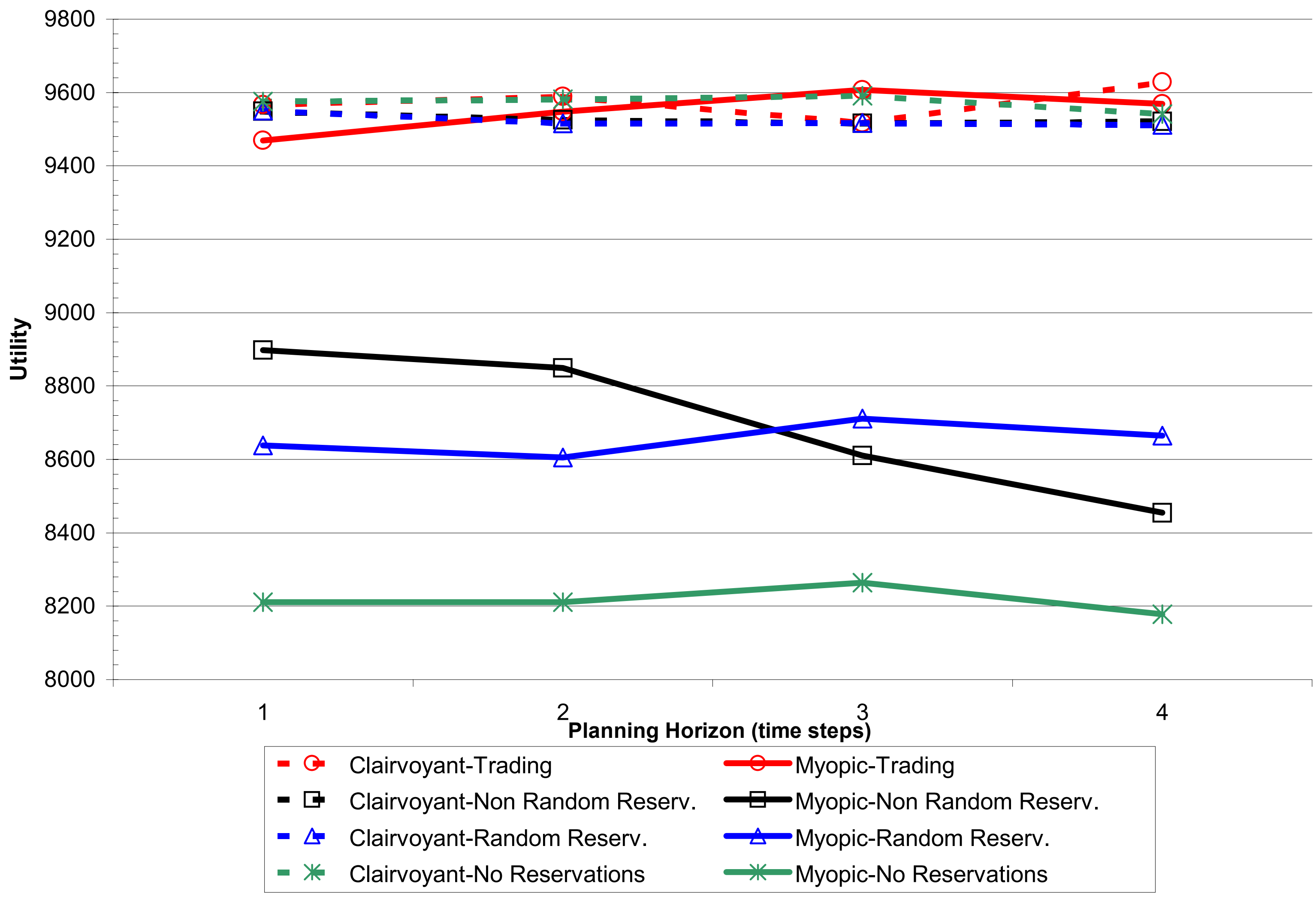

Figure 6.3 - First Problem Instance - Utility vs. Planning Horizon (20% Reservable Capacity)

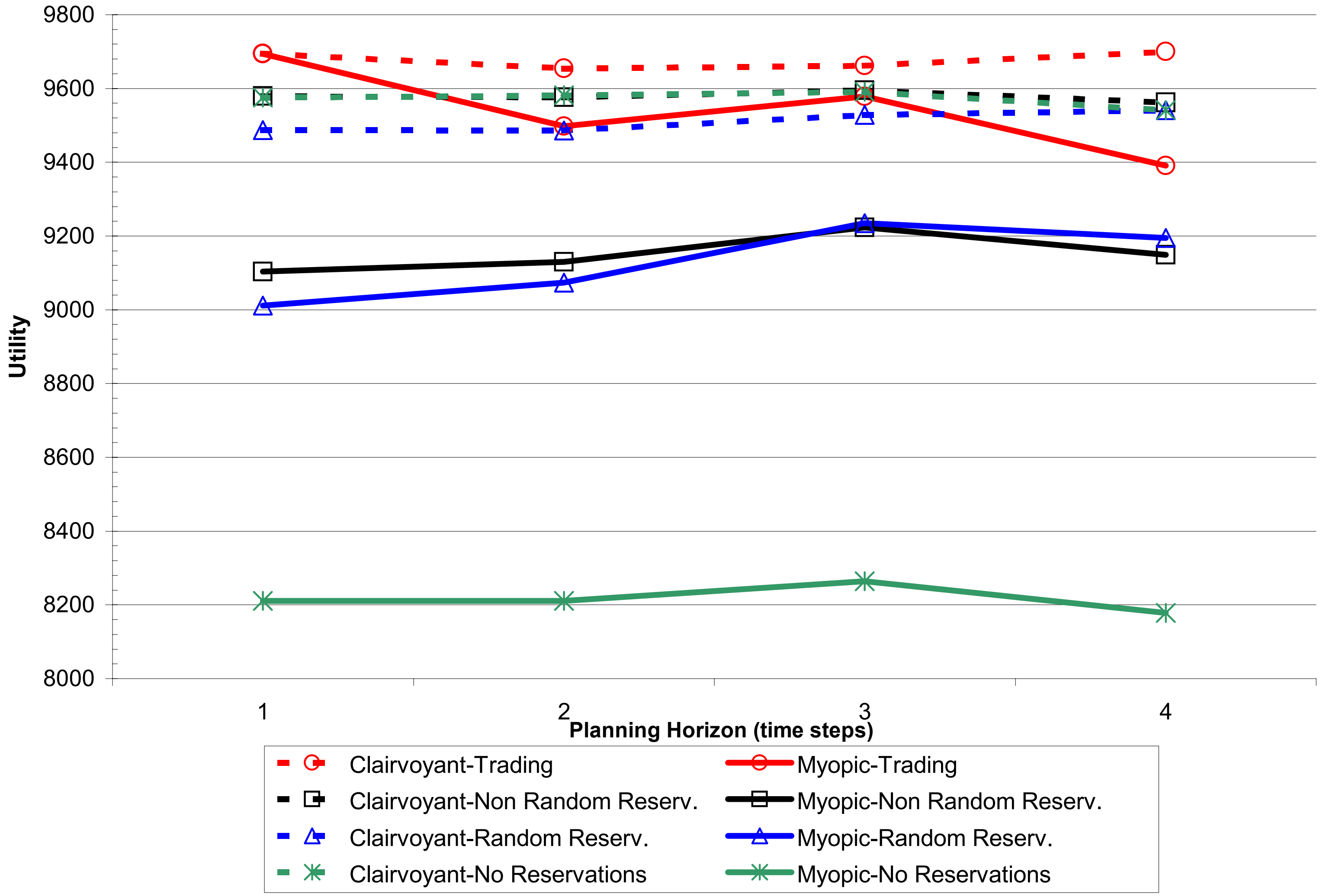

Figure 6.4 - First Problem Instance - Utility vs. Planning Horizon (40% Reservable Capacity)

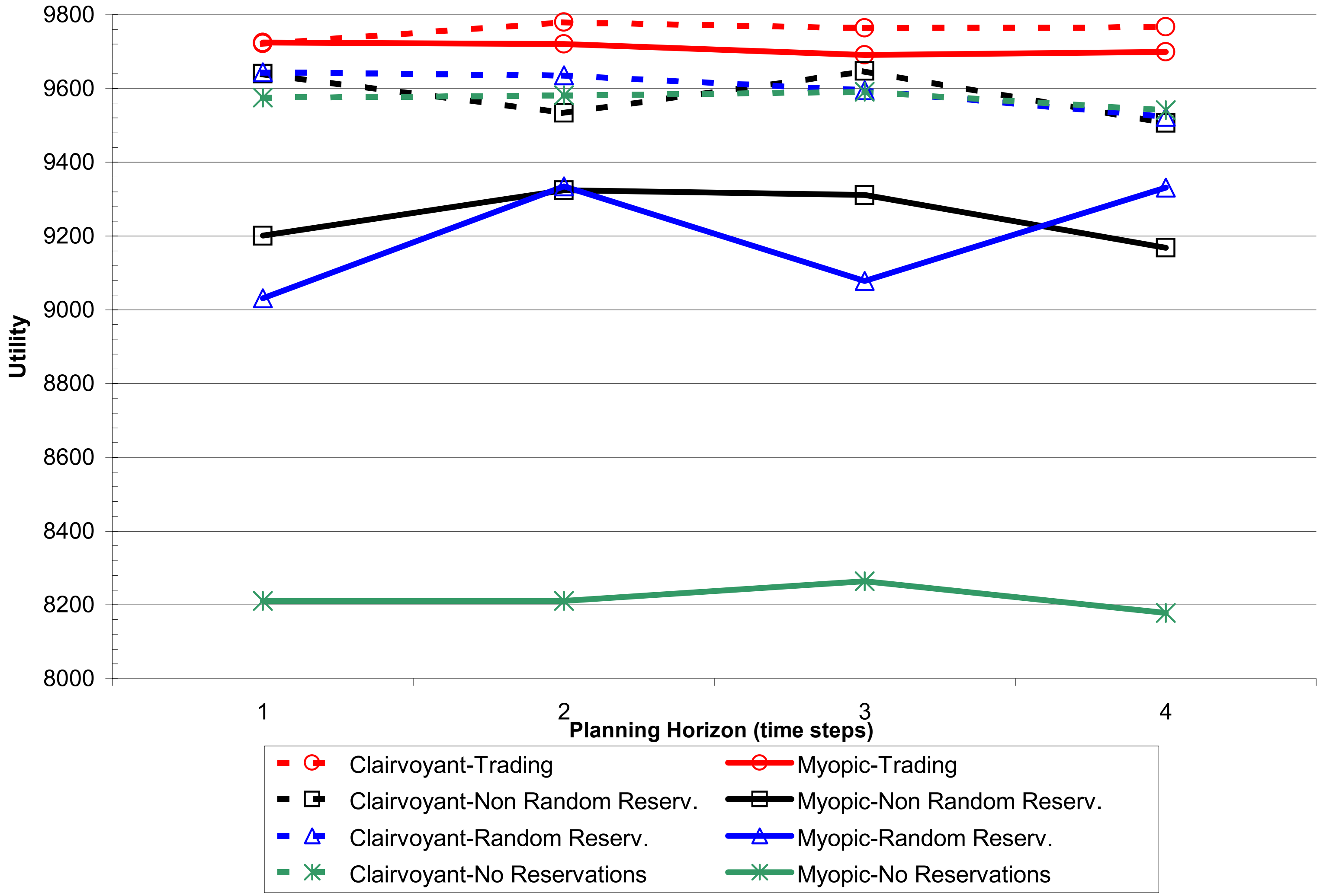

Figure 6.5 - First Problem Instance - Utility vs. Planning Horizon (70% Reservable Capacity)

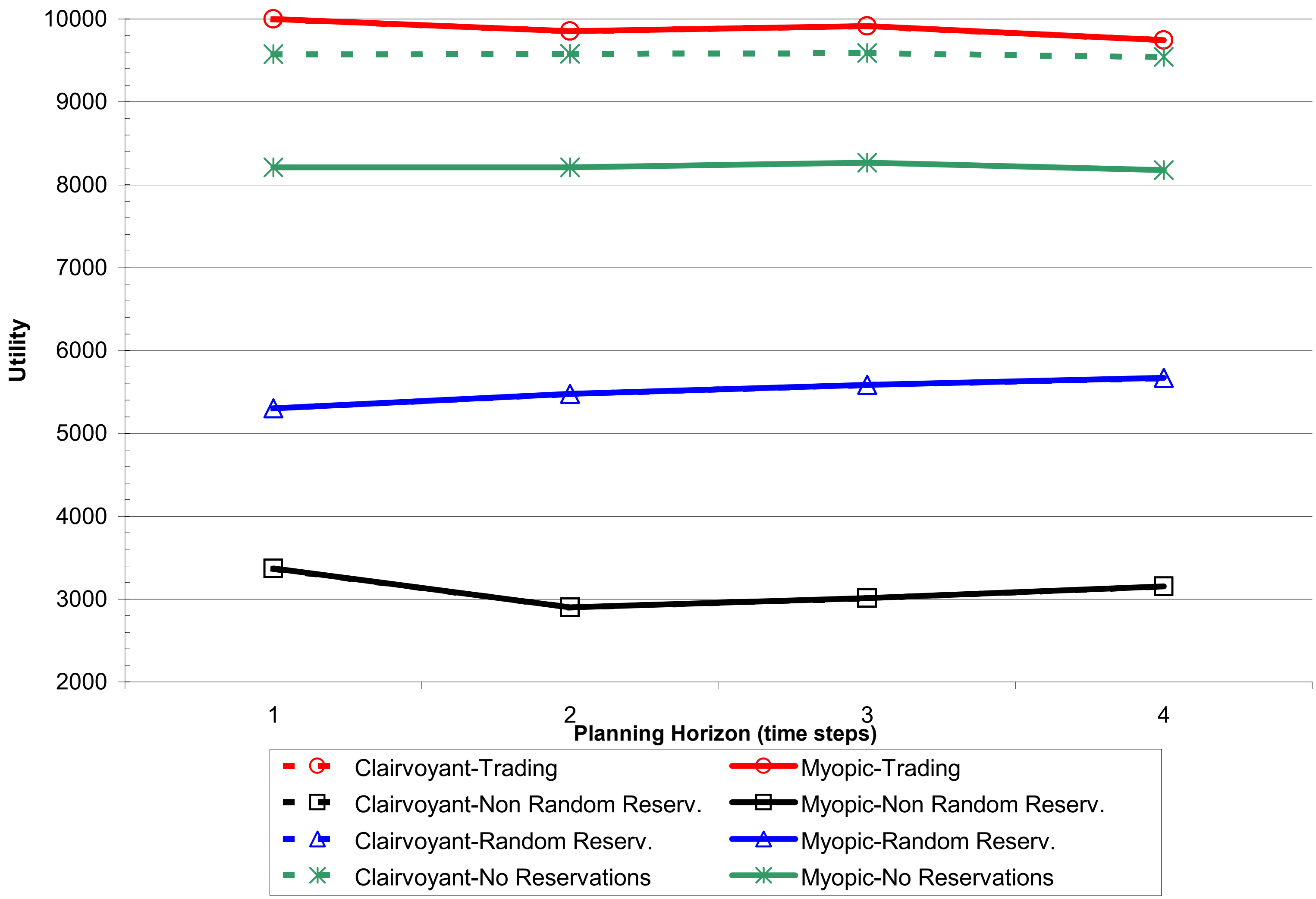

Figure 6.6 - First Problem Instance - Utility vs. Planning Horizon (100% Reservable Capacity)

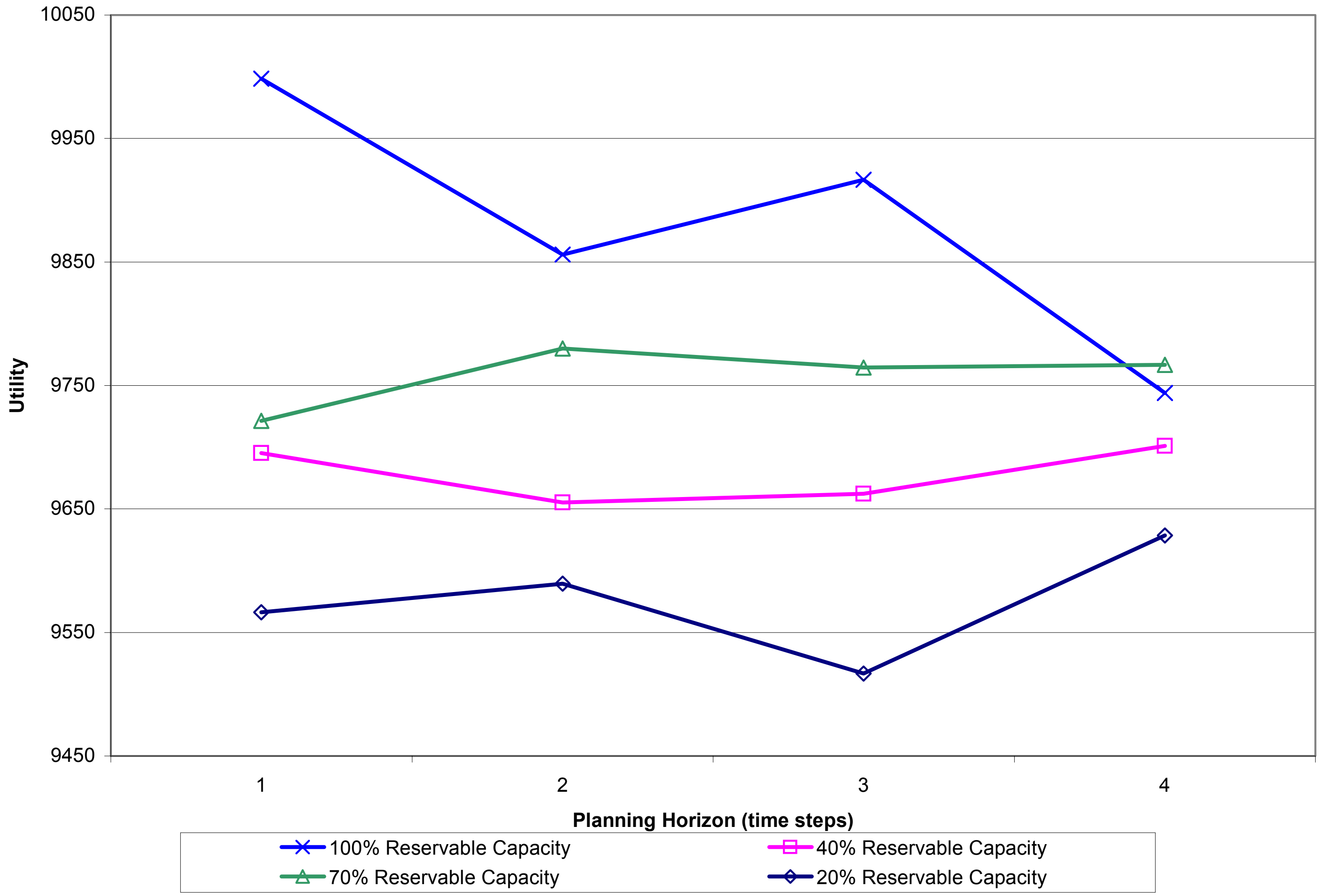

Figure 6.7 - First Problem Instance – Utility vs Planning Horizon with Clairvoyance - Trading

Table 6.5 - Results of Second Problem Instance - Myopic Simulation

Planning Horizon	Type of Simulation	20% Reservations		40% Reservations		70% Reservations		100% Reservations	
		Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used
1	No Reservations	10402.26		10402.26		10402.26		10402.26	
2	No Reservations	10437.19		10437.19		10437.19		10437.19	
3	No Reservations	10286.55		10286.55		10286.55		10286.55	
4	No Reservations	10027.29		10027.29		10027.29		10027.29	
1	Random Reservations - No Trading	10431.80	13.56	10570.14	10.96	10642.17	11.85	7348.95	40.49
2	Random Reservations - No Trading	10266.00	10.67	10305.90	10.44	10511.18	11.27	7469.09	40.93
3	Random Reservations - No Trading	10357.42	14.22	10390.34	9.85	10568.22	12.65	7467.52	41.98
4	Random Reservations - No Trading	10270.50	11.78	10100.01	11.26	10204.33	13.04	7450.88	43.53
1	Non-Random Reservations - No Trading	10410.24	11.33	10209.98	8.67	10384.98	8.50	5760.14	27.95
2	Non-Random Reservations - No Trading	10320.29	13.78	10348.17	9.41	10323.68	9.46	5942.72	29.81
3	Non-Random Reservations - No Trading	10217.85	14.00	10279.15	8.59	10336.67	7.23	5435.83	26.93
4	Non-Random Reservations - No Trading	10174.22	14.22	10063.79	7.48	10213.77	6.23	4919.84	25.98
1	Reservations - Trading	10741.81	62.89	11030.76	55.11	11529.27	51.58	11766.08	73.05
2	Reservations - Trading	10786.34	65.56	10979.27	57.26	11419.46	54.23	11743.94	73.23
3	Reservations - Trading	10568.59	59.44	10886.05	55.85	11272.77	52.73	11270.55	72.05
4	Reservations - Trading	10297.25	61.22	10691.80	56.22	11293.02	53.46	11062.74	72.09

Table 6.6 - Results of Second Problem Instance - Clairvoyant Simulation

Planning Horizon	Type of Simulation	Reservations = 20%		Reservations = 40%		Reservations = 70%		Reservations = 100%	
		Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used	Utility	% Rsv used
1	No Reservations	10448.12		10448.12		10448.12		10448.12	
2	No Reservations	10362.12		10362.12		10362.12		10362.12	
3	No Reservations	10200.96		10200.96		10200.96		10200.96	
4	No Reservations	9768.77		9768.77		9768.77		9768.77	
1	Random Reservations - No Trading	10424.38	9.78	10484.57	9.33	10611.28	10.12	7348.95	40.49
2	Random Reservations - No Trading	10244.54	10.89	10377.30	10.15	10511.38	10.81	7469.09	40.93
3	Random Reservations - No Trading	10129.26	11.33	10198.16	9.33	10323.84	10.54	7467.52	41.98
4	Random Reservations - No Trading	9880.39	11.33	9863.66	9.19	9997.61	11.38	7450.88	43.53
1	Non-Random Reservations - No Trading	10479.73	6.89	10312.38	8.00	10515.24	8.46	5760.14	27.95
2	Non-Random Reservations - No Trading	10334.75	12.44	10466.08	8.07	10445.50	7.96	5942.72	29.81
3	Non-Random Reservations - No Trading	10143.59	11.33	10156.00	7.11	10154.27	5.65	5435.83	26.93
4	Non-Random Reservations - No Trading	9882.44	16.22	9881.38	7.26	9854.54	5.38	4919.84	25.98
1	Reservations - Trading	10703.08	57.78	10985.30	52.52	11490.76	49.92	11766.08	73.05
2	Reservations - Trading	10661.30	60.44	10890.16	55.78	11468.04	49.19	11743.94	73.23
3	Reservations - Trading	10327.35	58.44	10805.78	54.74	11245.49	46.50	11270.55	72.05
4	Reservations - Trading	10055.71	65.11	10485.71	51.78	11077.76	46.50	11062.74	72.09

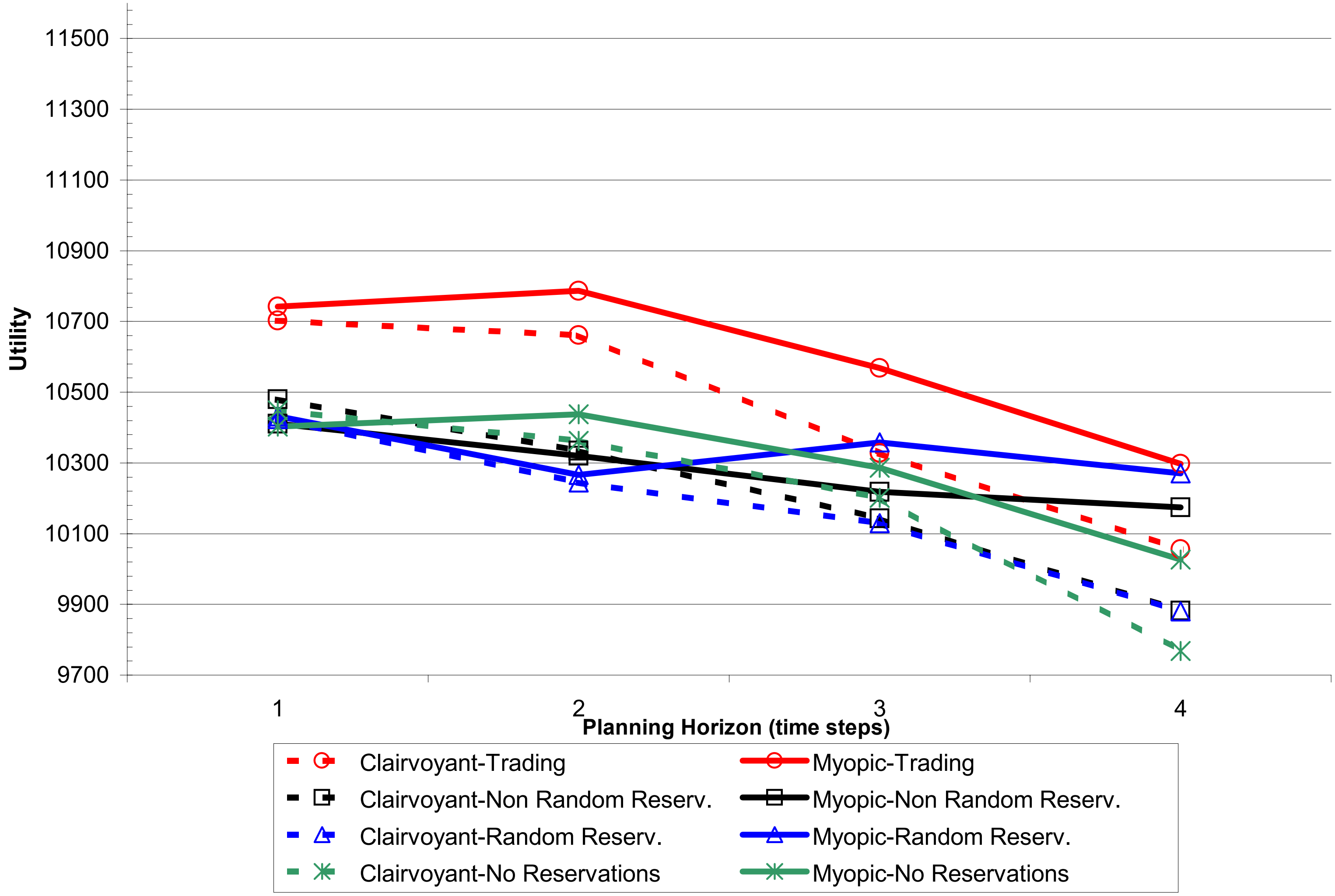

Figure 6.8 - Second Problem Instance - Utility vs. Planning Horizon (20% Reservable Capacity)

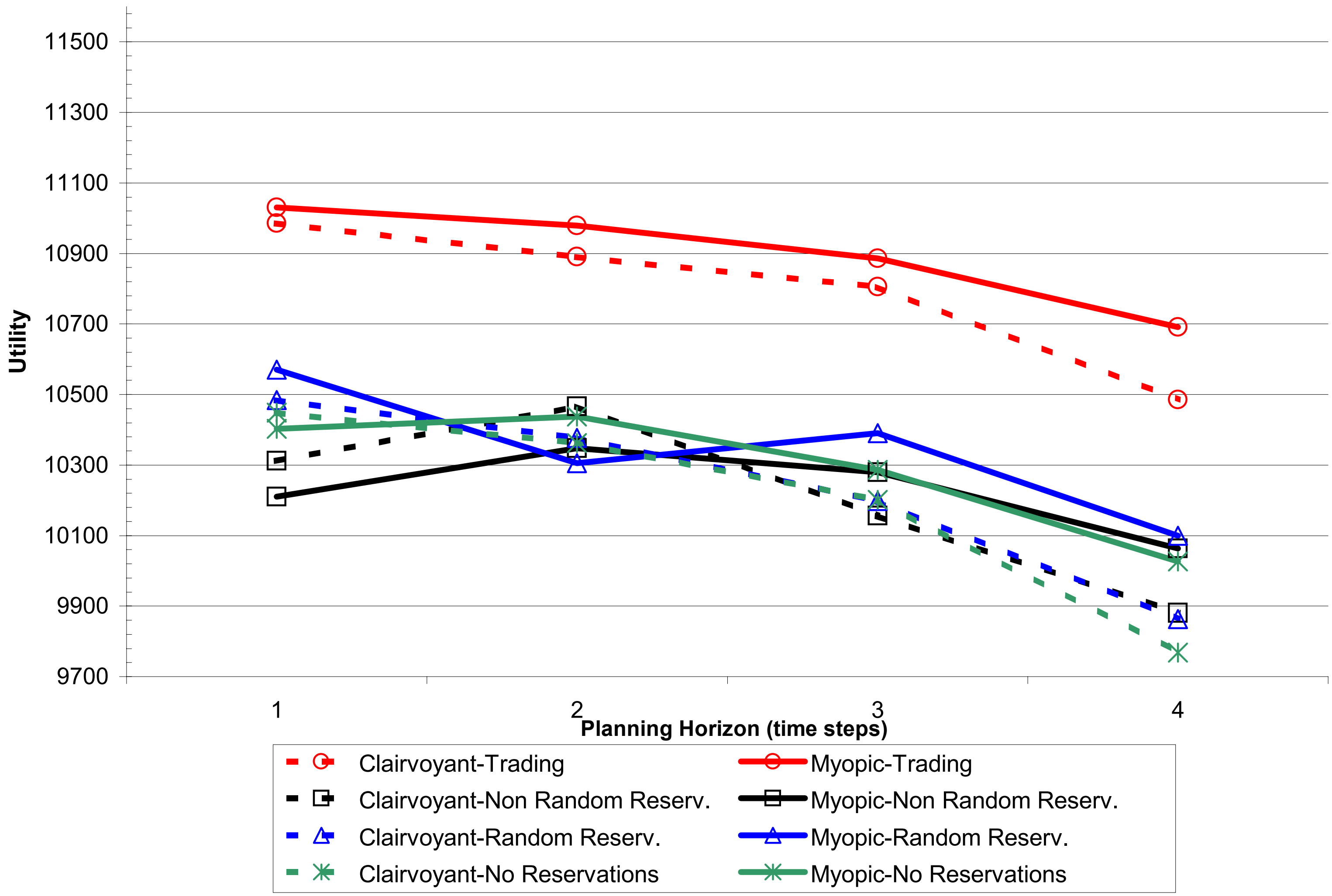

Figure 6.9 - Second Problem Instance - Utility vs. Planning Horizon (40% Reservable Capacity)

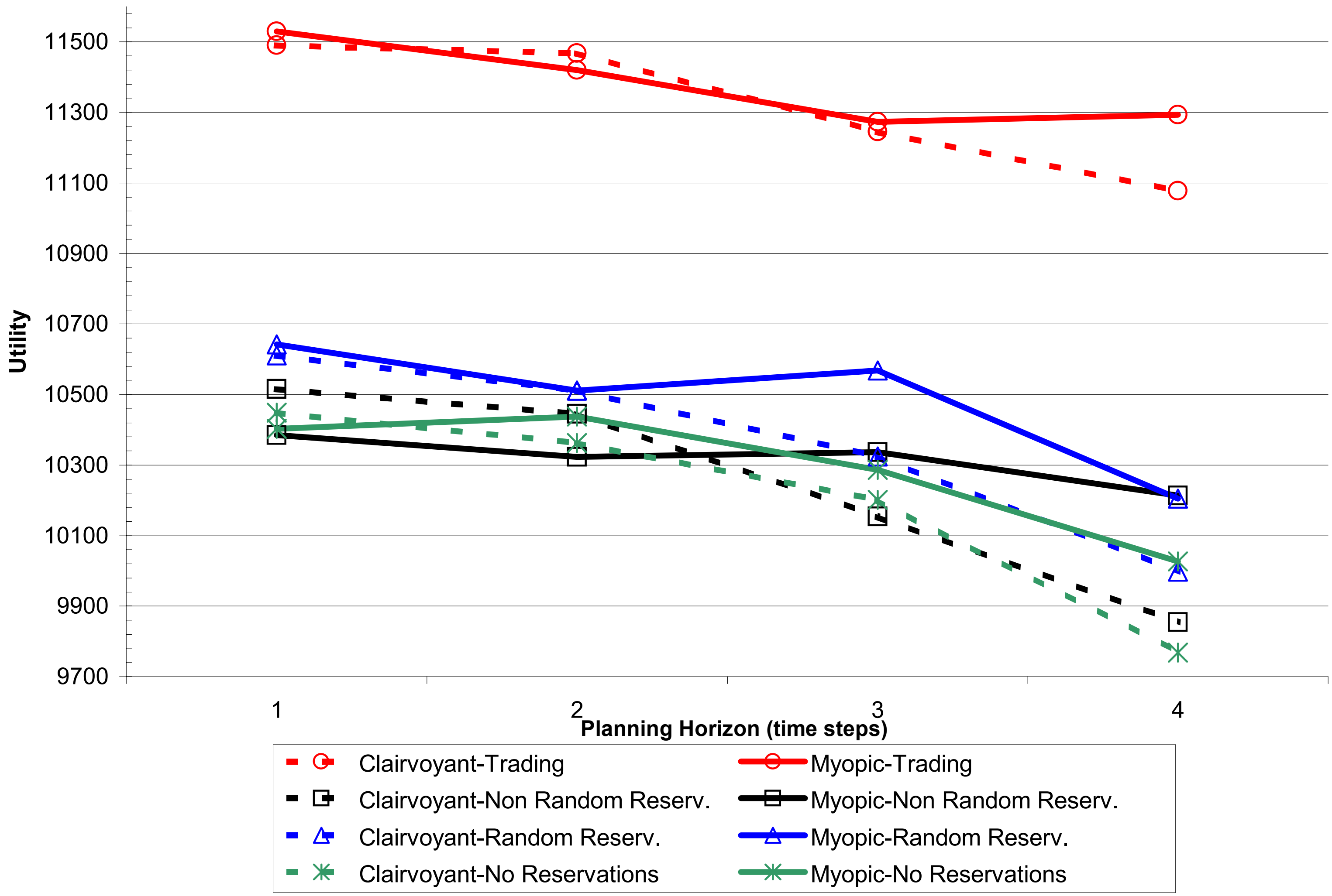

Figure 6.10 - Second Problem Instance - Utility vs. Planning Horizon (70% Reservable Capacity)

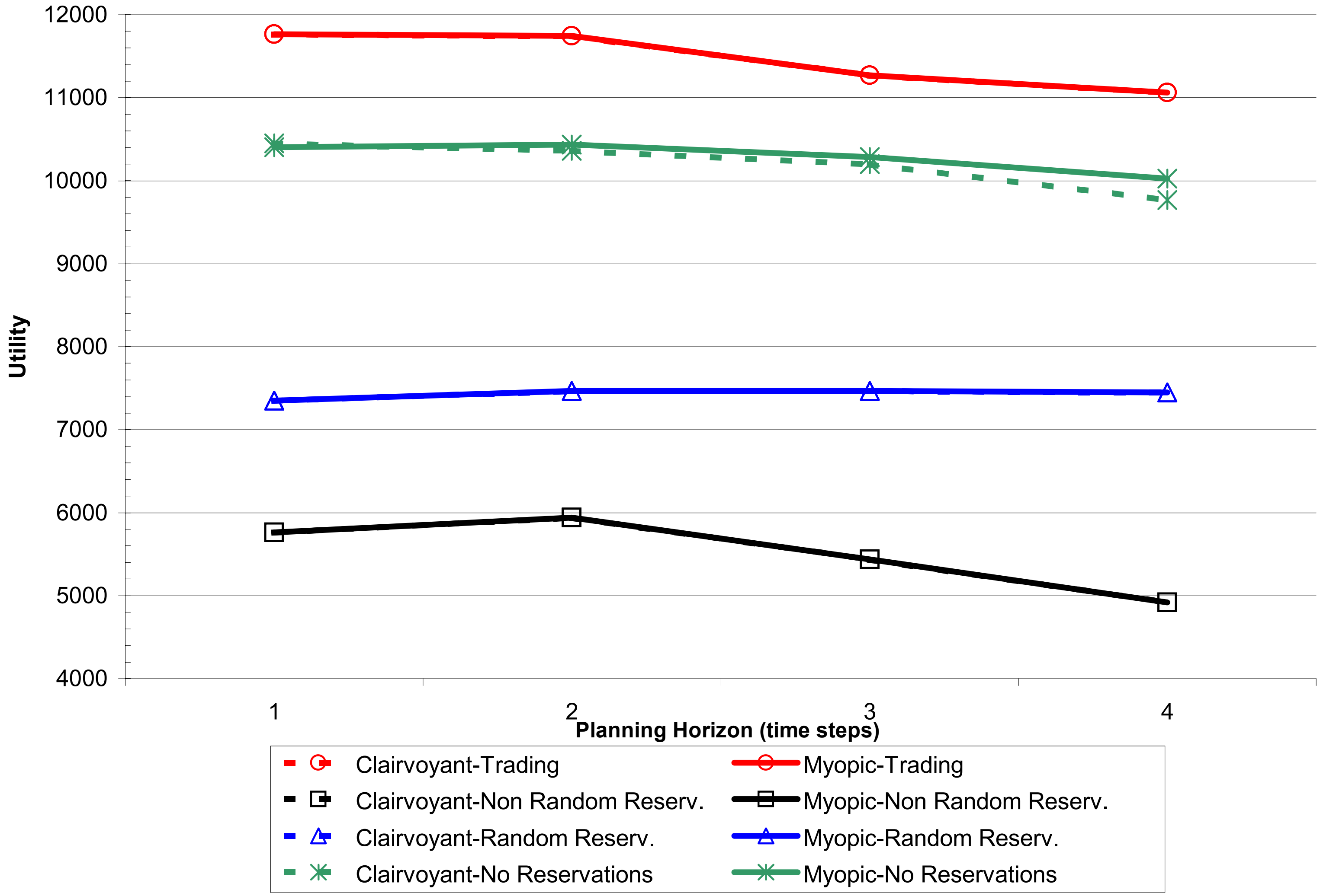

Figure 6.11 - Second Problem Instance - Utility vs. Planning Horizon (100% Reservable Capacity)

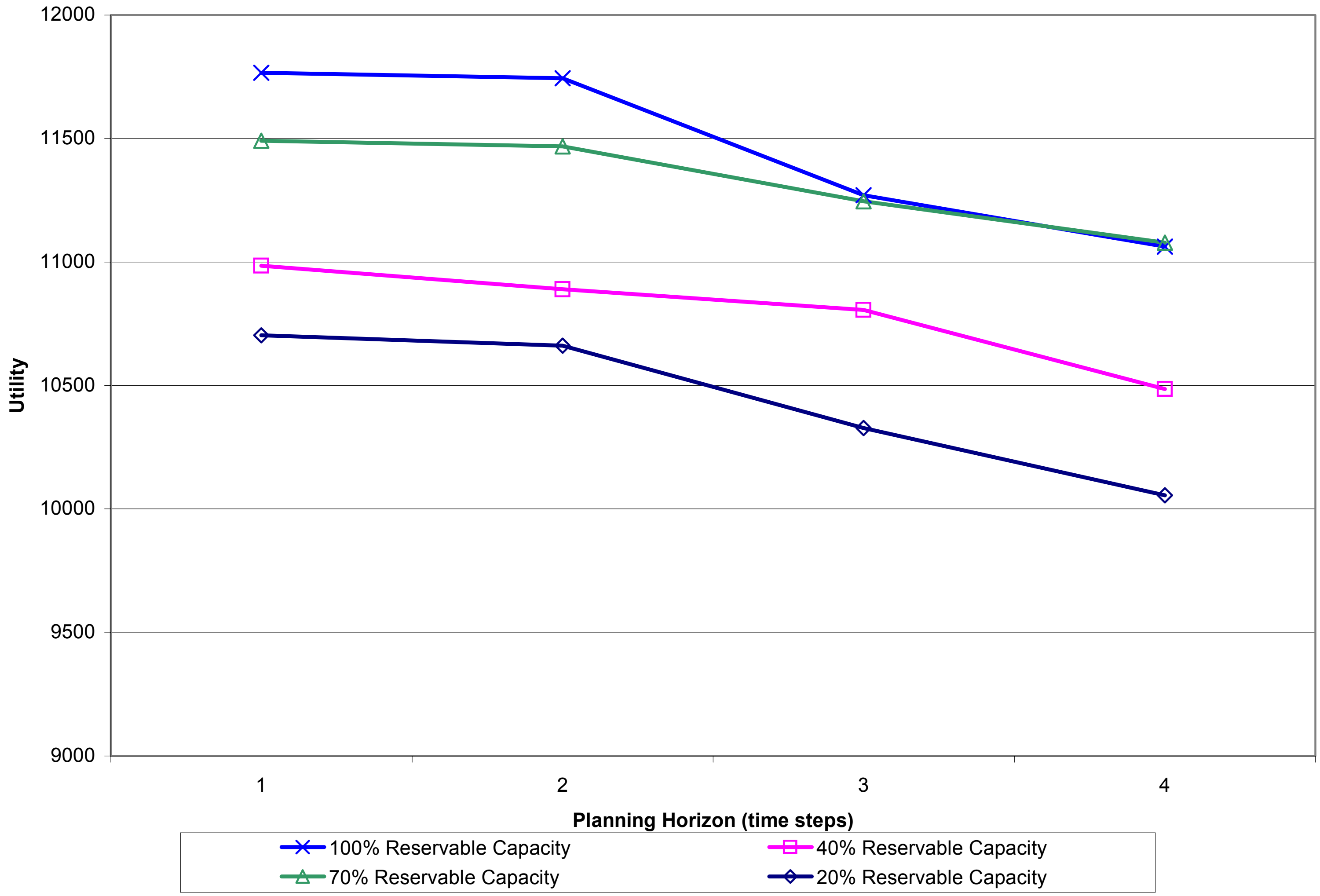

Figure 6.12 - Second Problem Instance – Utility vs Planning Horizon with Clairvoyance - Trading

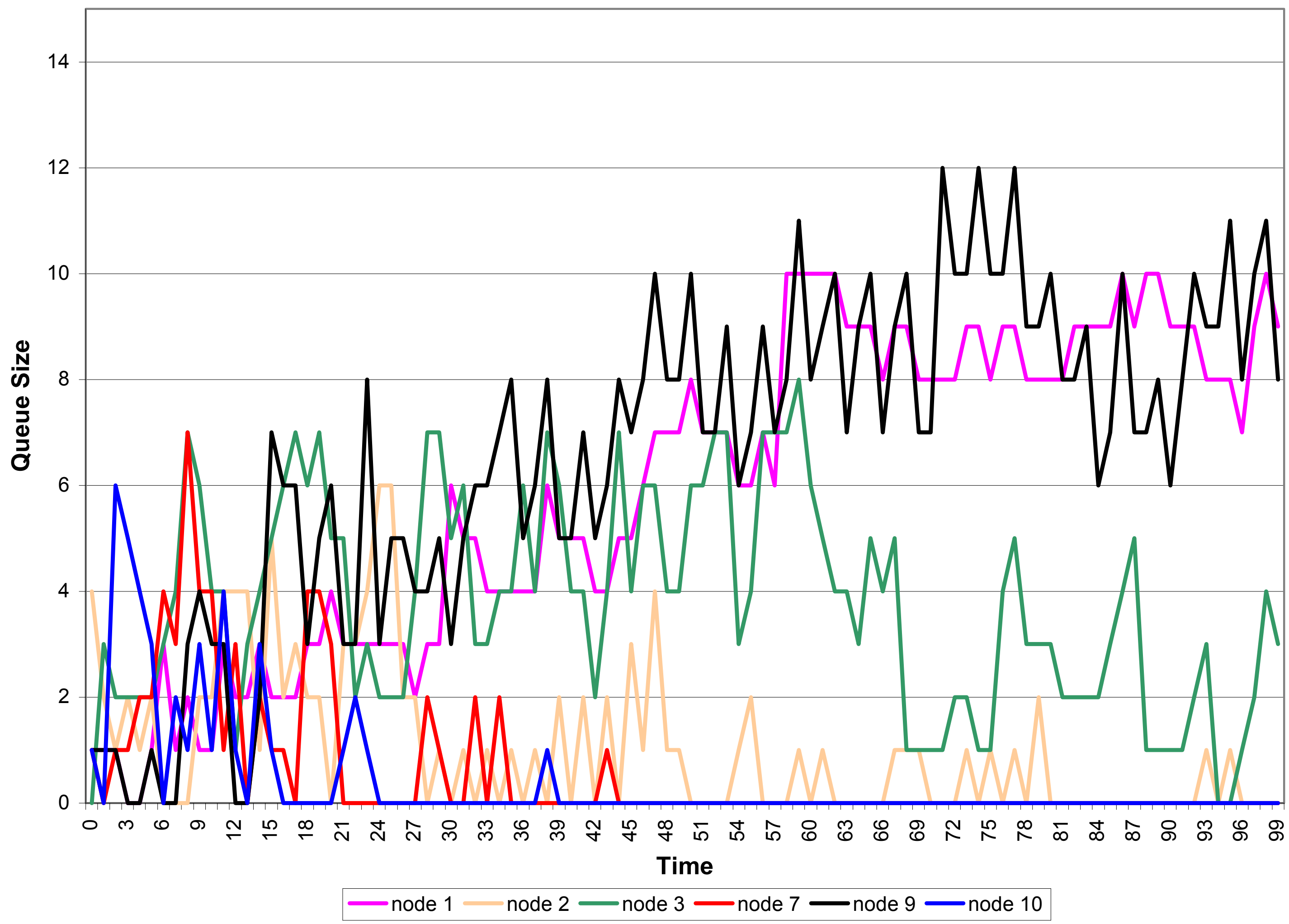

Figure 6.13 - First Problem Instance – Myopic - No Reservations - Queue Size Over Time

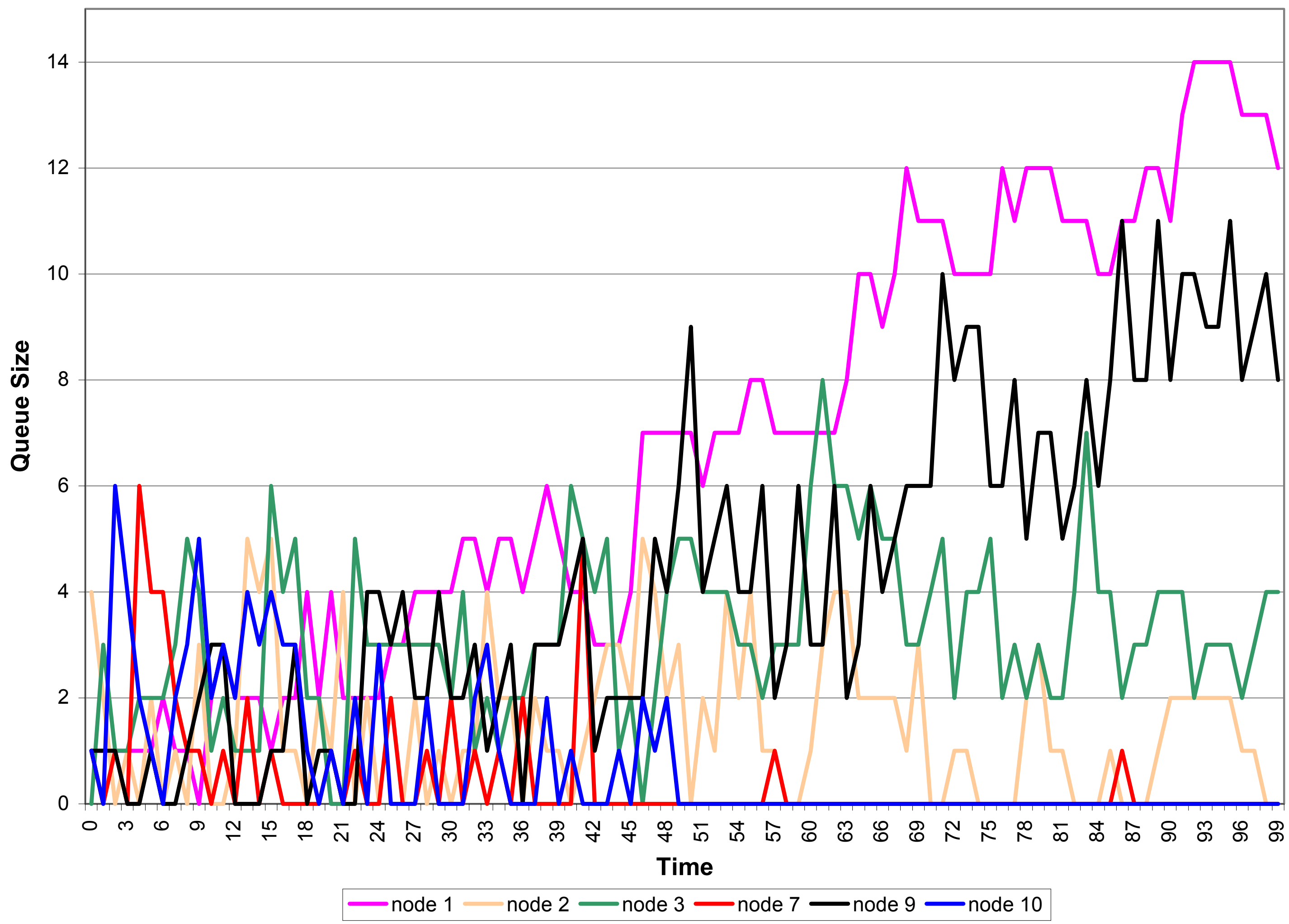

Figure 6.14 - First Problem Instance – Clairvoyant - 40% Reservable Capacity and Trading - Queue Size Over Time

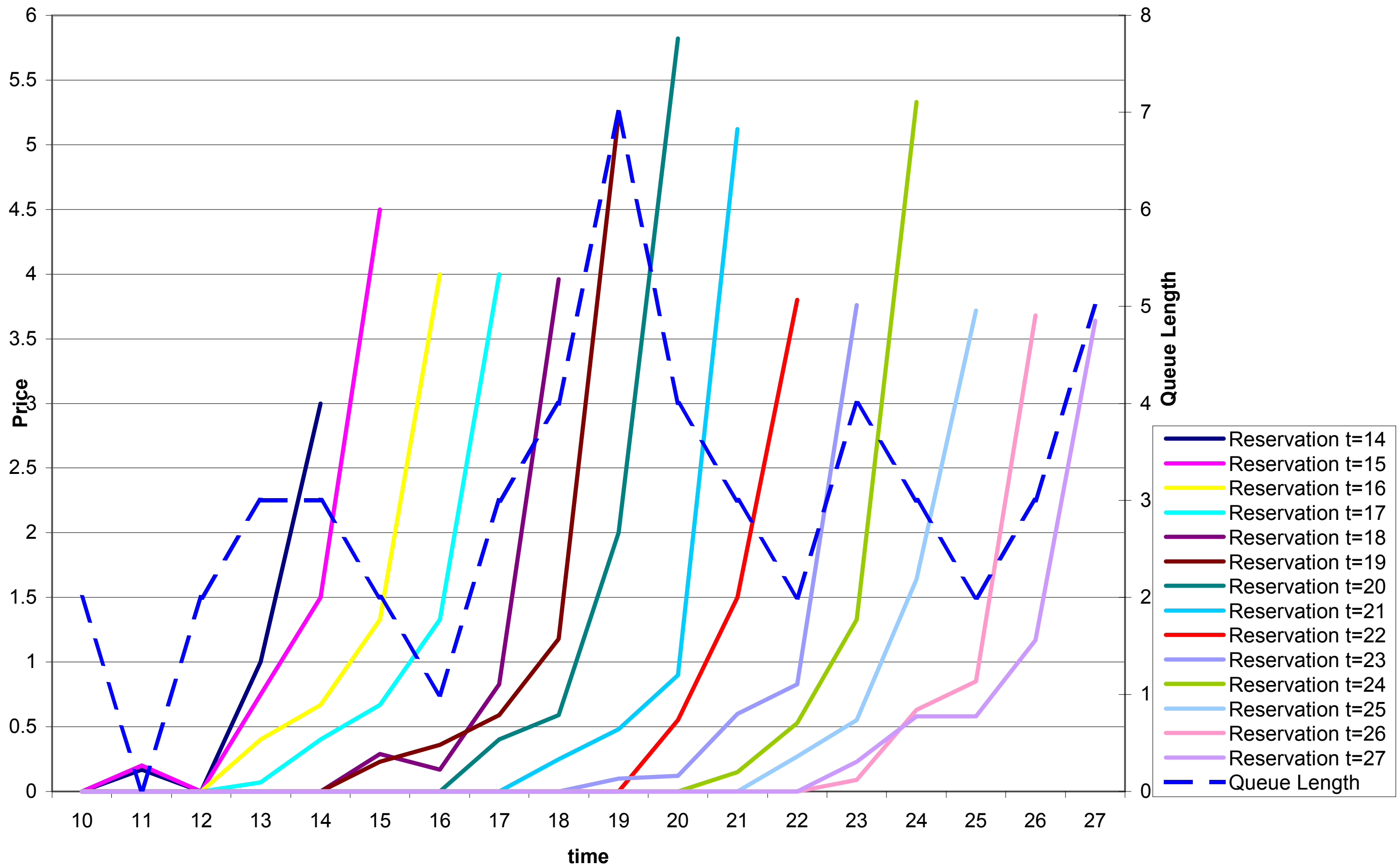

Figure 6.15 - Evolution of Prices - First Problem Instance–40 % Reservable Capacity–Clairvoyant–Planning Horizon 1 – Activity at Node 5

Chapter 7 Related Work

A great deal of recent work has studied software agents in electronic markets [2, 9]. However, to our knowledge, no one has modeled the types of environments that I have addressed in this thesis. The Electronic Elves project [1] is one project that studies the impact of mobile assistants that help workgroups coordinate their activities, but currently the project does not involve market interactions. The Trading Agent Competition [10] is a framework for studying trading strategies in a marketplace for travel resources. The agents in TAC, however, represent groups of people with travel preferences, but do not have an overly complex scheduling problem. Some work has been done on market-based scheduling [4, 11] but the constraints on the models differ in significant ways from the model presented here.

Chapter 8 Conclusions and Future Work

In this thesis I present a formal model of a common, multi-agent coordination problem in which agents are non-cooperative and resources are limited. Through simulation, I have experimented with the performance of the social system when the resources are reservable, and when agents can trade the reservation in a marketplace. I found that trading reservations improve the social welfare, but the benefits of better information and reservations alone depend on the configuration of the environment.

We plan to continue to extend the model. The random distribution of reservations should be replaced by some allocation mechanism that avoids impossible to use sets of reservations. The planning algorithm used here is very simple, and can be improved by integrating state of the art planning systems into the simulation (eg. partial-order planning). Investigating better market mechanisms like combinatorial auctions in this environment is also part of the future work. A more extensive analysis of the social welfare when changing parameters of the environment like the distance between nodes, number of agents and total capacity of the model is also in the research agenda. We also plan to study the ability of the enhanced system to accommodate plan deviations, as when a human user suddenly becomes interested in an activity that was not previously in the plan.

References

[1] Hans Chalupsky, et al. Electric Elves: Applying agent technology to support human organizations. *Proceedings of the Thirteenth Innovative Application of Artificial Intelligence Conference*. Seattle, WA, 51-58, 2001.

[2] Anthony Chavez and Pattie Maes. Kasbah: An agent marketplace for buying and selling goods. *Proceedings of First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, 1996.

[3] J. Q. Cheng and M. P. Wellman. The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes. *Computational Economics*, 12:1-24, 1998.

[4] Scott Clearwater. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific. 1995.

[5] Sven de Vries and Rakesh Vohra. Combinatorial Auctions: A survey. *INFORMS Journal on Computing*, to appear.

[6] R. W. Floyd, "Algorithm 97: Shortest Path", *Communications of the ACM* 5(6), 345, June 1962.

[7] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[8] R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699-738, 1987.

[9] Michael P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1, 1–23, 1993.

[10] Michael P. Wellman, et al. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2), 43-51, Mar.-Apr. 2001.

[11] Michael P. Wellman, William E. Walsh, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35(1-2), 271-303, 2001.

[12] Peter R. Wurman, William E. Walsh, and Michael P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.