Decentralized and Dynamic Home Health Care Resource Scheduling Using an Agent-Based Model

Zhijie Xie

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

July, 2016

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Zhijie Xie

Entitled: Decentralized and Dynamic Home Health Care Resource Scheduling Using an Agent-Based Model

And submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

Dr. Anjali Awasthi Chair

Dr. Onur Kuzgunkaya (MIE) External Examiner

Dr. Jia Yuan Yu Internal Examiner

Dr. Chun Wang Supervisor

Approved by

Chair of Department or Graduate Program Director

2016

Dr. Amir Asif, Dean

Faculty of Engineering and Computer Science

ABSTRACT

Decentralized and Dynamic Home Health Care Resource Scheduling Using an Agent-Based Model

Zhijie Xie

The purpose of this thesis is to design an agent-based scheduling system, simulated in a dynamic environment that will reduce home healthcare service costs. The study focuses on situations where a health care agency needs to assign home visits amongst a group of independent healthcare practitioners. Each practitioner has different skill sets, time constraints, and cost structures, given the nature, time and location of each home visit. Each expects reasonable payment commensurate with their skill levels as well as the costs incurred. The healthcare agency in turn needs all planned visits performed by qualified practitioners while minimizing overall service costs. Decisions about scheduling are made both before and during the scheduling period, requiring the health care agency to respond to unexpected situations based on the latest scheduling information.

This problem is examined in a multi-agent system environment where practitioners are modeled as self-interested agents. The study first analyzes the problem for insights into the combinatorial nature of such a problem occurring in a centralized environment, then discusses the decentralized and dynamic challenges. An iterated bidding mechanism is designed as the negotiation protocol for the system. The effectiveness of this system is evaluated through a computational study, with results showing the proposed multi-agent scheduling system is able to compute high quality schedules in the decentralized home healthcare environment. Following this, the system is also implemented in a simulation model that can accommodate unexpected

situations. We presents different simulation scenarios which illustrate the process of how the system dynamically schedules incoming visits, and cost reduction can be observed from the results.

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisor Professor Chun Wang the patient guidance, encouragement, and advice he has provided throughout my studies. I have been extremely lucky to have a supervisor who cared about both my academic career and professional career. His mentorship has contributed to my academic success, while his advice has strongly influenced my career choice. I am grateful to have had the opportunity to work with him.

I must also express my gratitude to my family for their enduring support throughout my research, and to my friends and colleagues in the lab, without whose encouragement I undoubtedly could not have successfully achieved this milestone.

Finally, I would like to thank ENCS of Concordia University, not only for providing the funding which allowed me to undertake this research, but also for giving me the opportunity to pursue my Master's degree in such an outstanding academic environment.

Table of Contents

ABSTRACT iii
ACKNOWLEDGEMENTS v
Chapter 1 Introduction 1
1.1 Background and Motivation 1
1.2 Scope and Approach 2
1.3 Outline of the Thesis 4
Chapter 2 Literature Review 6
2.1 Home Health Care Resource Allocation 6
2.2 Centralized, Decentralized and Dynamic Approaches 10
2.2.1 Centralized Approaches 10
2.2.2 Decentralized Approaches 12
2.2.3 Dynamic Approaches 15
2.3 Other Resource Allocation Problems in Home Health Care 18
2.4 Summary 19
Chapter 3 Home Visit Scheduling Problem 21
3.1 Mathematical Model 21
3.2 The Challenge of Decentralized Environments 24
3.3 The Challenge of Dynamic Environments 25
Chapter 4 Decentralized Scheduling for HVS 28
4.1 Multi-Agent System Design 28
4.1.1 System Architecture 28
4.1.2 Negotiation Protocol 29
4.1.3 System Initialization 31
4.2 Practitioner Agent's Local Decision Making 32
4.3 Health Care Agency Local Decision Making 33
4.4 A Worked Example 34
4.5 Computational Study 42
4.5.1 Design of Testing Data 42
4.5.2 Experimental Results 44
4.5.3 Implementation Consideration 45
4.6 Summary 46
Chapter 5 Dynamic Scheduling for HVS 47
5.1 Dynamic Scheduling through Periodic Repair 47

5.2 Evaluating the Dynamic Scheduling Algorithm through Simulation 50
5.3 Simulation Model Design 51
5.3.1 Model Structure 53
5.3.2 Running Environment 54
5.3.3 State Chart of Practitioner Agent 56
5.3.4 State Chart of Health Care Agency 58
5.4 Simulation Model Implementation 60
5.4.1 Feasibility Checking 60
5.4.2 Distance Calculation 61
5.4.3 Feasible Package Generation 62
5.4.4 Bid Creation 63
5.4.5 Termination Checking 64
5.4.6 Iterative Bidding 64
5.4.7 Optimization Model for Winner Determination Problem 65
5.5 Simulation Scenarios and Results 66
5.6 Summary 72
Chapter 6 Conclusions and Future Work 73
References 75

List of Tables

Table 1 Symbols used in this thesis 22
Table 2 Summary of Visits 36
Table 3 Travel Time between Visits (hours) 37
Table 4 Practitioners' Feasible Packages and Corresponding Cost and Payment 38
Table 5 Practitioner Information 40
Table 6 Submitted Bids, Provisional Allocation, Payment and Cost at Each Round of Bidding 41
Table 7 Schedule Generation Algorithm 43
Table 8 Cost Estimation Algorithm 44
Table 9 Practitioner's Cost and Health Care Agency's Payment at Different Group 45
Table 10 Dynamic Scheduling Algorithm 50
Table 11 Feasibility Checking Algorithm 61
Table 12 Distance Calculation Algorithm 62
Table 13 Feasible Package Generation Algorithm 62
Table 14 Bid Creation Algorithm 63
Table 15 Termination Checking Algorithm 64
Table 16 Iterative Bidding Algorithm 65
Table 17 Planned Visits in Run 1 67
Table 18 Profile of Practitioners 68
Table 19 Initial Schedules 68
Table 20 Dynamic Scheduling Result in Run 1 68
Table 21 Final Schedules in Run 1 68
Table 22 New Planned Visits in Run 2 69
Table 23 Final Schedules in Run 2 69
Table 24 Dynamic Scheduling Result in Run 2 69
Table 25 New Planned Visits in Run 3 70
Table 26 Final Schedules in Run 3 70
Table 27 Dynamic Scheduling Result in Run 3 71
Table 28 New added Practitioners in Run 4 71
Table 29 Dynamic Allocation Result in Run 4 71
Table 30 Final Schedules in Run 4 72
Table 31 Summary of Simulation Results 72

List of Figures

Figure 1 Relationship between the TSP, VRP and variants 8
Figure 2 FIPA Iterated Contract Net Interaction Protocol 15
Figure 3 System Architecture 29
Figure 4 Negotiation Protocol 30
Figure 5 Dynamic Scheduling Flowchart 48
Figure 6 Dynamic Scheduling Workflow 49
Figure 7 Class Diagram of Simulation Model 52
Figure 8 Sequence Diagram of Simulation Model 53
Figure 9 Running Environment of Simulation Model 54
Figure 10 Practitioner Agent State Chart 55
Figure 11 Health Care Agency State Chart 59
Figure 12 Optimization Model for Winner Determination in Simulation 66

Chapter 1 Introduction

Home health care plays an increasingly important role in the overall health care system, a situation that is complicated by the fact that more and more practitioners are self-employed nowadays. Contracting with health care agencies, who in turn represent either health insurers or government agencies, these home care professionals provide health services in their patients' homes. Scheduling information is unevenly distributed across a complex environment made up of autonomous agencies and individuals, making it difficult for any agency to construct an accurate schedule as no single party has a full view of the entire homecare landscape. Moreover, agencies typically seek to provide high quality service by making the schedule as flexible as possible to accommodate unexpected situations. Although considerable effort has been put towards resolving the challenges of home health care scheduling, few researchers have proposed solutions that take into consideration the uniquely decentralized environment and dynamic characteristics in the problem.

1.1 Background and Motivation

Home health care is increasingly seen as complementary to conventional hospitalization, due mostly to limited hospital capacity and rising health care costs. Recent years have seen frequent discussions on how to contain the growth of health care costs, with public healthcare systems and government agencies under tremendous financial pressure to deliver services in a cheaper, more efficient manner. According to the Canadian Institute for Health Information (CIHI), total health care expenditures increased by around 100% over the last ten years, much higher than the growth of either Canada's gross domestic product (GDP) or population over the same period [1]. Recent reports show that a total of $141 billion was spent on health care in 2014, with the average

Canadian family contributing $11,735 in taxes towards public health insurance in 2015. It has also been reported in the United States that a decade of rising health care costs from 1999 to 2009 wiped out real income gains for the average US family, while, similar to Canada, containing the escalation of costs in the US healthcare system has become a difficult challenge due to the complexity of the system's interacting parts [2].

As an integral part of many health care systems, home health care aims to provide treatment for illness or injury inside the patient's residence, playing an important role in helping patients recover and regain their independence. The concept of home health care is a distinct type of medical service, usually distinguished from other non-medical care. Licensed medical personnel, such as registered nurses, licensed practical nurses, or therapists, usually provide such services. Another concept mentioned frequently is home care, which normally includes both medical care and non-medical care. This thesis uses 'home health care' in the broader sense that also includes home care.

A wider range of health care services or treatments can presently be provided at the convenience of the clients' home, including doctor care, nursing care, laboratory testing, or other basic assistance. Transportation expenses involved in traveling between clients' homes, as well as idle time wasted between visits must be reduced, however, in order to contain home health care service costs. A growing proportion of health care practitioners in health care systems around the world tend to be self-employed rather than hospital or other health care institution employees. Increasing numbers of independent practitioners changes the home health care environment, increasing the complexity of conventional resource allocation problems in this field. Meanwhile, unexpected situations must still be accommodated in this very different health care environment.

1.2 Scope and Approach

This research is concerned with reducing service costs in home health care. It focuses on situations in which a home health care agency contracts with a group of independent health care practitioners to provide health care services to its clients. The home care agency can be a representative of a health care insurer or a government agency. The main tasks of the agency include collecting visit requests from clients and scheduling practitioners to complete the visits at the client's home. In practice, the fast changing environment and the desire to provide high quality service makes it difficult to follow a set schedule. Unexpected events occur, such as new service requests, cancellations, or urgent cases. Visiting requests need to be collected dynamically over the course of the schedule. To health care practitioners, the home health care agency is the customer who pays for these home visits. The practitioners each have different skill sets, time constraints, client preferences, and payment requirements. On the other hand, the patients have their own medical requirements, language preferences, etc. While accommodating the preferences of both parties, the home health care agency is also obligated to schedule practitioners according to incoming visit requests either before the start of a new schedule period or during the schedule period, while at the same time seeking to reduce costs as much as possible.

In a highly decentralized environment, service providers have sufficient autonomy to control their own schedules. Their relationship with the health care agency is a contractual one; they are not employees of the health care agency. Practitioners each have different costs of covering a schedule (a bundle of visits) based on their work time, travel costs, and even client preferences. A particular schedule might not be feasible to a practitioner in a given day due to scheduling conflicts or negative utilities. Health care agencies need to schedule home visits in a way that practitioners' time constraints are satisfied and all home visits are covered by the right practitioners, while at the same time the overall cost are minimized.

Before the schedule takes effect, an initial plan for each practitioner is generated. Schedules may be revised due to various changes, e.g., new visit requests, cancellations, or changes in the timing of visits. We model the scheduling problem in a multi-agent setting, consisting of a health care agency and multiple practitioner agents. The practitioners are modeled as self-interested agents, and the cost of covering a schedule is their private information, not known to the health care agency. A practitioner's primary objective is to maximize their payoff, which is the difference between the payment received and the costs incurred in covering a schedule. On the other hand, the objective of the health care agency is to minimize overall payments to practitioners for covering all visiting requests, given that the cost to practitioners are not known to the agency. Unexpected situations are modeled as discrete time events, which reflect new visit requests, cancellations, or visit time updates. Cancellations from patients are non-negotiable, while visit time updates are considered a two-step procedure, consisting of both a cancellation and a new visit request. During the schedule period, the main concern is to accommodate new visit requests according to practitioner schedules. The conflict of interest between the health care agency and the practitioners calls for game theoretic modeling and solutions to the recourse-scheduling problem in home health care. In addition, simulation modeling is needed due to the dynamic behaviors involved in the problem.

This thesis develops an agent-based scheduling system to support cost reduction in home health care by assigning cost effective schedules to practitioners. It also develops a simulation model for accommodating the dynamic situations. Our first contribution is the design of a decentralized scheduling algorithm, implemented by a bidding mechanism. This also serves as the negotiation protocol of an agent-based system, enabling both the agency and the practitioner to construct efficient home visit schedules through an automated multilateral negotiation. We also present decision-making tools for both agencies and practitioners in the system. Our other contribution is to enhance the algorithm to further accommodate dynamic scheduling events. We implement the enhanced algorithm in a simulation model, which simulates the scheduling process in a dynamic environment. In this model, the schedule performance process is also simulated as part of the decision-making activity of practitioners.

1.3 Outline of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 reviews resource allocation problems in the home health care field, analyzing the characteristics of the scheduling problem in particular, followed by a review of models developed to deal with such issues. The reviewed solutions include centralized, decentralized, and dynamic approaches. Other resource allocation problems are also briefly reviewed. Chapter 3 describes the Home Visit Scheduling problem, and formulates a centralized model to deal with it, while also discussing the decentralized and dynamic challenges of the problem. Chapter 4 presents an agent-based system and the scheduling approach. An iterated negotiation protocol is designed, combined with a bidding framework to facilitate cooperation between agents. The bidding mechanism is proposed as the solution for decentralized scheduling. In chapter 5 we enhance the approach to further accommodate dynamic situations and events affecting scheduling. A simulation model is developed to demonstrate the dynamic scheduling process. Chapter 6 concludes the thesis and discusses future research directions.

Chapter 2 Literature Review

This chapter briefly reviews relevant work in the home health care field. The first section introduces the home health care resource allocation problem and its characteristics. Following this, various scheduling approaches are reviewed from a more general perspective in section 2. The other resource allocation problems of home health care are briefly reviewed in section 3. Finally, we summarize these approaches and point out the distinct direction of our study.

2.1 Home Health Care Resource Allocation

The terms ‘home health care’ and ‘home care’ are not always clearly distinguished in the literature. In some cases, they are used interchangeably, e.g. in [3] and [4]. In the field of home health care operations management, the main issues are classified according to three levels, which are districting, assigning, and scheduling. Districting problems occur when an agency must group operators and patients into clusters, named districts, according to relevant criteria. Assigning problems involve assigning each district to practitioners in an impartial way. Resource scheduling problems are perhaps the most important issue, and as such have attracted more research attention than the other two. Such problems aim to assign practitioners to visit under a set of constraints. The dimensioning problem of home health care is considered a higher-level issue, requiring researchers to determine the number of practitioners necessary to satisfy care demands as a whole [3]. Our review focuses on resource scheduling problems, with other problems only briefly reviewed at the end of this chapter.

In home health care operations management literature, considerable efforts have been devoted to resource scheduling. The general home care resource-scheduling problem has the nature of both a rostering problem and a routing problem. In [5], Castillo-Salazar et al. treated the problems of workforce rostering and routing separately. Other literature built the model combining the rostering aspect with routing [6]. From the rostering perspective, the problem is to assign practitioners to visits with a set of hard and soft constrains. A similar problem that has been extensively studied is the 'nurse rostering problem' (NRP), alternatively called the 'nurse-scheduling problem'. Solutions to the NRP aim to generate a functional roster for nurses while satisfying the constraints of both nurses and the hospital. A comprehensive survey of NRP solutions can be found in [7]. Additionally, this problem also has the nature of a routing problem, seeking to traverse the assigned visits [8]. Much of the early literature considers home health care resource scheduling problems as extensions of the Traveling Salesman Problem (TSP) or the Vehicle Routing Problem with Time Window (VRPTW), e.g., [9] [10] and [11], to name just a few.

TSP is one of the most intensively studied optimization problems in the literature. It requires us to determine the optimal route for a salesman to visit a given number of cities exactly once from a depot and then return back to the depot when finished, minimizing travelling costs. The VRP requires us to find the optimal routes for a set of vehicles to deliver goods to a set of customers, starting and ending at a single depot. VRPTW is a variant of the more general Vehicle Routing Problem (VRP), with several other variants also being studied in the literature. Fig. 1 illustrates the relationship among TSP, VRP, and its variants [12], though further review is unfortunately beyond the scope of this paper. Other problems share similar characteristics to the home health care resource-scheduling problem, such as scheduling technicians performing repairs or security guards performing rounds [5]. The common characteristic of such problems is a scenario where personnel must travel to different locations and complete tasks without concern for time or in different time windows. Some other research focuses specifically on different schedule terms (short term, mid-term and long term) [13, 14].

Models proposed for home health care resource scheduling in the literature usually try to achieve one or more objectives under certain constraints. These objectives are normally setup to minimize service costs, minimize travel time or distance, or balance workloads among practitioners, depending on specific requirements. The cost is considered as one of the most important objectives in this problem, e.g., in [8] and [15]. A typical cost function includes service costs, travel costs, and normally also involves penalty costs that occur when there are some

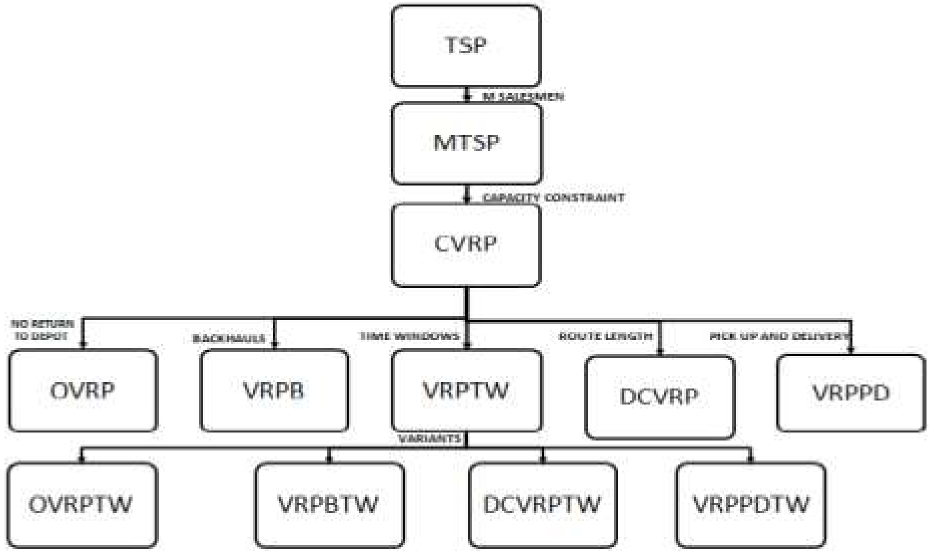

Figure 1 Relationship between the TSP, VRP and variants

preference violations. Practitioner and patient preferences are mostly linked to service levels and/or quality, both of which are considered in the constraints. The most common constraints involved in such problems include time, transportation modality, skills/qualifications, start and end locations, connected activities, shift types, and visit types. Most models categorize these as either hard or soft constraints, and almost all studies took the time constraint into account. Examples of time constraints in home health care include travelling times or waiting times between visits, service times, visit appointment times, and preferred working times of practitioners. Some of these are considered hard constraints, which cannot be violated, while others are treated as soft constraints, where violation is allowed with a violation penalty. For example, the start time of a visit is a hard constraint, while the waiting before the start of a visit is considered a soft constraint and as such is allowed [8]. Transportation costs are typically different for each practitioner, with transportation modality referring to the practitioner's particular mode of transport, e.g., car, bicycle, walking or public transportation, which in some extent reflects different travel costs.

We recognize two types of models based on the characteristics of the home health care scheduling problem and the features of the model used in the literature analyzed above. One type of model is based on VRP or TSP (a comprehensive review can be found in [12]). A few examples are listed as follows. Based on the generalization of VRPTW, Mutingi designed several methodologies from either a variant of existing algorithms or from novel ones to improve the solution efficiency under different assumptions [9, 16]. Akjiratikarl et al. proposed heuristic approaches based on VRPTW as well [17]. Another example is a model generalized from multi-depot VRPTW with multiple objectives [18].

Eveborn et al. [19] developed another typical model, formulating the home care problem as a set-partitioning model, minimizing costs and solving the problem through repeated matching algorithms. There are two constraints in their basic model: 1) each person selects exactly one schedule, and 2) each visit must be included in a chosen schedule. A decision support system called the Laps Care system was developed for Swedish health care organizations based on their model. In their study, schedules are generated a few days in advance. Given the service type and duration, the health provider allocates qualified staff to each visit. The health provider is divided into different areas where they provide services, such that the size of scheduling is decreased to a practical level. The preferences of patients are also considered in their research, while manual scheduling is employed to handle dynamic events. A typical hybrid model, reflecting both rostering and routing aspects, is presented as a core optimization model in some research [8]. The objective here is to minimize travel costs and maximize the satisfaction of both patients and nurses. The constraints in their model include covering all jobs exactly once, job start time constraints, taking the service time and travel time into consideration, the upper and lower bounds of working time, respecting nurses' individual schedules, and qualification requirements.

Despite the previously mentioned model type dimensions, most approaches in the literature opt to build centralized models, and propose either exact, heuristic or hybrid methods for their solutions. The centralized model helps in recognizing the essential features of the problem. However, the home health care resource-scheduling problem has decidedly decentralized and dynamic characteristics that are rarely touched on in the literature. As we described in the previous chapter, practitioners decide their own feasible schedules based on constraints and costs, which are only known to themselves. Practitioners are self-interested agents with no control over others. The approaches proposed by researchers cannot be directly used as solutions to the scheduling problem raised in a decentralized environment. Furthermore, their models cannot be used for any prolonged period of time due to the regular occurrence of unexpected situations in practice. Changes inevitably occur during the course of a schedule period, and such changes must be accommodated by health care agencies in order to provide high quality service. A proper rescheduling method is therefore needed to improve the service quality in a dynamic environment.

2.2 Centralized, Decentralized and Dynamic Approaches

In order to develop a decent solution for the problem, we review the approaches from centralized, decentralized and dynamic perspective separately in this section. We found that most of the literature focuses only on centralized approaches to home health care scheduling, though we also review decentralized approaches in related literature. We also review dynamic approaches for outpatient scheduling, appointment scheduling, and dynamic scheduling in manufacturing fields.

2.2.1 Centralized Approaches

Begur et al. presented a spatial decision support system used by home health care providers in the US [10]. The system combines stand-alone GIS software with a scheduling algorithm, as well as a user-friendly interface. They also introduced a heuristic approach consisting of several steps to generate and enhance routes. Such a system saved travel costs and improved workload balance for the health care organization.

Cheng et al. proposed a heuristic algorithm to solve a mixed-integer programming (MIP) model to minimize work time [11]. In their model, full-time workers who are paid for both shift work and overtime are distinguished from part-timers who are paid by the hour. The qualification requirement is simplified to a binary value, which indicates whether a practitioner is qualified or not. The algorithm has two phases, creating routes by way of a random greedy algorithm in the first phase, and improving the solution in the second phase. The objective is to minimize the working hours and overtime of full-time nurses and the working hours of part-time nurses.

Hiermann et al. gave special consideration to the modality of routes, and proposed a two-stage approach to solve real world multimodal home-healthcare scheduling problems [20]. The objective here is to minimize both travel time and constraint violations. In the first stage of the approach, initial solutions were generated using either constraint programming techniques or a random procedure. In the second stage, the initial solutions were iteratively improved by applying metaheuristic algorithms. They tested the performance of different algorithms in the second phase, such as variable neighborhood search, memetic algorithm, scatter search, and simulated annealing hyper-heuristic.

Mutingi and Mbohwa described a simulated evolution algorithm based on fuzzy set theory [21], which they used to minimize the time window violation and maximize the workload balance, at the same time maximizing the clustering efficiency of the schedule [22]. They also presented a simulated metamorphosis algorithm that considers the problem in both a fuzzy environment [23] and group genetic algorithms, with the objective of minimizing travel costs and time window violation penalty costs [9]. The genetic algorithms in the latter research are designed based on an extension of VRPTW. Thomsen solves the home care problem using a tabu search algorithm to minimize travel time and maximize the number of visits in his Master's thesis [24]. Bertels and Fahle presented an optimization model to minimize the travel cost and maximize the satisfaction of both patients and nurses [8]. The model is abstracted from the optimization module in an industrial prototype. The proposed approach combines linear programming, constraint programming, and heuristics to find the optimal solution.

Eveborn et al. formulated the home health care problem as a set partitioning model, providing flexible architecture, and solved this problem through repeated matching algorithms [19]. Some other methods based on the set partitioning model have also been used to solve the home health care scheduling problem, such as the visit clustering methods presented in [18] and [25], particle swarm optimization methods presented in [16] and [17], and neighborhood search methods presented in [15, 26-28].

In addition to heuristic methods, exact methods are also studied in the literature. Kergosien et al. modeled the home health care scheduling problem as an extension of multiple traveling salesmen with time window constraints, solving it exactly with a CPLEX solver [29]. Redjem et al. suggested an MIP for the home health care scheduling problem, with both precedence and coordination constraints [30]. They compared the complexity of different temporal dependencies and the number of care activities per caregiver ratio between the two models based on the Traveling Salesman and Resources Constrained Project Scheduling Problems.

Yalcindag et al. reviewed those studies, which consider the home health care resource-scheduling problem as a TSP or VRP in the home health care context [6]. They analyzed the relationship among the variants of these problems and differentiated them according to two dimensions, each of which has two categories. The first dimension is static or dynamic, while the second one is deterministic or probabilistic/stochastic. They reviewed studies based on four categories, which are general characteristics, modeling characteristics, network characteristics, and data characteristics, pointing out that the solutions to problems in the first dimension are the same, while they are all applicable to dynamic case. The challenge is that the time dimension increases the complexity of dynamic scheduling. Mutingi and Mbohwa also did a comprehensive review of home health care staff scheduling [31]. They described commonly used models, different objective functions, and classified constraints (time-based, demand-based, and preference-based) in home health care scheduling.

2.2.2 Decentralized Approaches

In our setting, practitioners are independent from the health care agency, and as such they make decisions according to their availabilities, preferences and the payoffs of taking a particular schedule. It should therefore be considered a decentralized scheduling problem in a multi-agent system, which calls for economic-based solution models. An auction and bidding mechanism is an effective economic-based model of procuring products and services in a market setting. There are several types of auction used in the resource-scheduling field. Single item auctions are useful when there is only a single bidding item at any given time. Generalized Vickrey Auction (GVA) is used in small size scheduling problems due to its computational complexities. Iterative bundle auctions are one of the implementations of GVA that does not require the complete and exact valuation information of agents, also offering the potential to support dynamic scheduling.

The application of auctions and bidding mechanisms to the domain of home health care scheduling is a relatively new research direction. While a few previous studies in home health care used agent-based approaches, they rarely involved economic models. Bajo et al. built large open multi-agent systems expanded from FIPA architecture, which consisted of Service Facilitator, Organization Management System and Platform Kernel [32]. Such architecture is applied to the home care area in supervising and monitoring the patients at home. There are four kinds of roles interacting in home health care implementation: patient, doctor, family, and provider. In this model, entities in different roles communicated with each other based on complicated automatic reasoning and planning mechanisms. Because home health care operations require effective communication within a distributed environment, Baho et al. believe that agent-oriented methodologies provide better mechanisms for modeling distributed, inter-operable, and secure systems by taking social and organizational considerations into account. However, the proposed solution focuses on improving the intelligence system rather than resource scheduling.

However, auction mechanisms have also been applied to many general resource-scheduling problems. In the hospital context, Grano et al. proposed a two-stage approach consisting of a sealed bid auction and schedule completion to solve the NRP problem [33]. The first stage in their auction includes a bidding round and a winner determination considering both nurse preferences and hospital constraints, with unassigned shifts handled in the second phase. In their approach, nurses are given a fixed number of points as their "budget", and submit a bid package consisting of shifts with pre-allocated points. The winners are then awarded shifts and the remaining shifts are assigned to those nurses whose schedule is not fully allocated through an additional optimization model. The researchers used real-world data to conduct their experiments, producing a better solution when compared with those generated by manually self-scheduling.

An earlier example is the combinatorial auction mechanism proposed in [34]. Four versions of this mechanism are applied to the job shop-scheduling problem. A typical application is the model developed for grid resource allocation [35]. Moreover, Lau et al. designed a combinatorial auction mechanism for distributed resource allocation and applied such a mechanism to solving a scheduling problem in a container terminal [36]. Dargahi et al. proposed an iterative bidding framework to schedule service requests in a software service domain [37]. A customer's value on a schedule reflects the preference of completion time, which is private information. The customers are considered to be self-interested, motivated to maximize their own payoff. The computing capacity is allocated to service requests, such that the profitability of the service provider is maximized. The experimental results show that this framework achieves a higher efficiency compared with a first-come-first-served scheduling policy. It was also observed that a larger epsilon requires more information revelation but less computing time.

In the mass customization field, Wang and Dargahi proposed a combinatorial iterative bidding framework to solve the problem of service customization under capacity constraints in service customization settings [38]. The objective is to maximize the overall customer value, which reflects their social welfare, under the given service capacity. In their research, the proposed framework is applied to travel package customization, in which it serves as a multilateral negotiation protocol between agents. The agents follow a myopic best-response bidding strategy, and the incentives of both seller and customer is studied form game theory perspective. The experiment is then compared with a first-come-first-serve allocation policy, which shows the former policy has a higher performance. Mehrizi and Wang also applied iterative combinatorial auction to carrier collaboration [39]. In logistic service areas, the carriers work collaboratively to take orders. The authors proposed a descending bidding framework with the objective of minimizing carriers' overall costs. The experiment's results show a high level of performance compared to optimal solutions, and the increasing competition leads to decreasing procurement costs. The researchers also conducted an application of reverse iterative combinatorial auction to optimize the resource utilization rate of peer-to-peer communication [40].

In a decentralized environment, a negotiation protocol is needed to let different parties communicate with each other following some predefined criteria. Contract Net Protocol is commonly used in decentralized scheduling. It was first proposed by Smith [41], and then included in the Foundation for Intelligent Physical Agents (FIPA) standards. The basic Contract Net is a decent protocol for distributing one task among participants. One initiator and multiple participants are involved in such a protocol. Knable et al. extended this to Contract Net with Confirmation Protocol (CNCP), in order to deal with concurrent multiple tasks in a setting that involved multiple initiators [42]. The proposed protocol tackles the issue that early commitment can lead to suboptimal solutions when basic Contract Net protocol is used. The idea of their solution is to postpone the participant's commitment time by adding an iterative process of asking and answering one-by-one until one participant agrees to take the task from the initiator. FIPA Iterated Contract Net Interaction Protocol is extended from the basic FIPA Contract Net, allowing multiple rounds of iterative bidding [43]. As illustrated in Fig. 2, the initiator may issue a revised call for proposal to iterate the process. With such a protocol, the receivers of the revised call for proposal are limited to the accepted participants. The termination conditions of the iteration are either 1) the initiator refuses all proposals and does not call for a proposal, or 2)

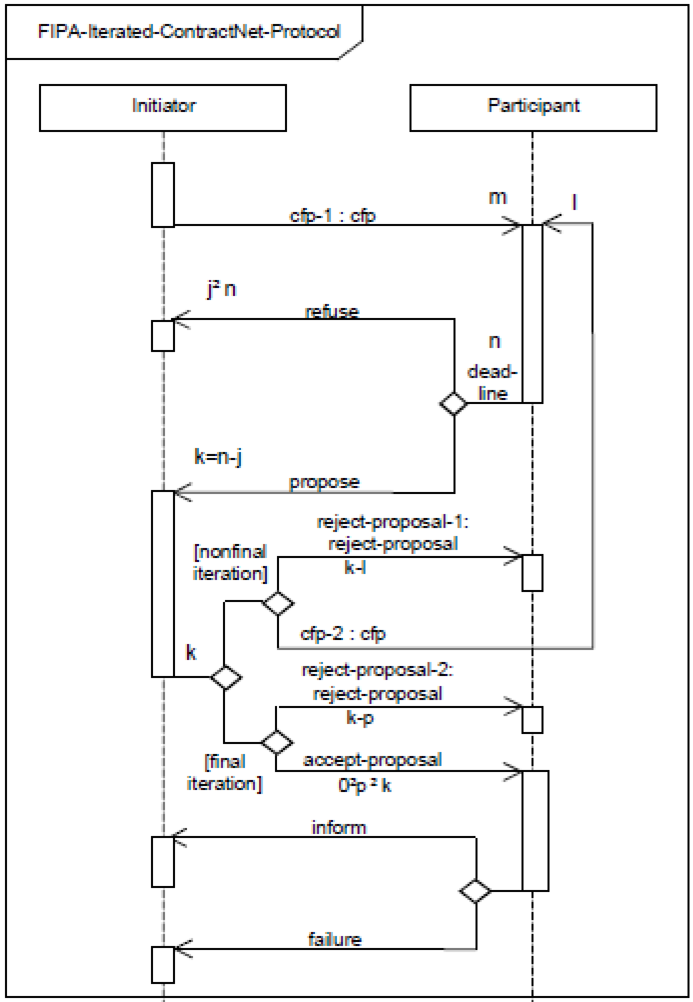

Figure 2 FIPA Iterated Contract Net Interaction Protocol

one or more bids are accepted, or 3) no participants are willing to bid. One application of the Iterated Contract Net protocol is a health monitoring system designed for NASA, in which the agents negotiate to determine the task allocation and call in a workflow engine to actually perform the tasks [44]. We obtained ideas from this work when developing our own simulation model.

2.2.3 Dynamic Approaches

While dynamic home health care scheduling is mentioned in some works, few researchers focus on such a problem. However, some related home care research has studied rescheduling from a human-computer interaction perspective. Lanzarone et al. considered the scheduling problem as one of synchronous collaboration activities conducted by a group of planners in a realistic environment [45]. A weekly plan was generated in advance, with planners rescheduling collaboratively on a daily basis through the designed interaction system (a tangible, multi-touch table), by which the issues involved are more understandable for planners. In [46], a novel data representation method was proposed as a visualization framework to enhance collaboration between planners. In situations where automatic optimization algorithms are unable to generate desired solutions or respond effectively to dynamic events, the proposed method supports human analysis in a way that enhances perception, cognition, and information exchange.

Few researchers specifically study rescheduling problems in health care contexts. Some complementary tools of traditional methodologies have therefore been introduced to support human decision-making in rescheduling. Godin and Wang proposed an agent-based, dynamic distributed scheduling approach, which allocates available diagnostic service timeslots to outpatients through an iterative bidding procedure [47]. There are three agents included in the model, representing clinic schedulers, directory facilitators (DF), and patients in multi-agent systems architecture. Contract Net protocol is used for coordination between the Diagnostic Services (DS) agent and the patient agent. The DS agent sends out a call for proposal to patient agents registered with a DF agent based on a set of available timeslots, with patient agents then sending bids on a set of timeslot bundles back to the DS agent. The DS agent then computes and assigns the timeslots, updates the existing schedule, and starts over with another call for proposals. This procedure continues until no timeslots are available or no patient agents are willing to take on new timeslots. The DS agent and patient agent each have their own decision-making model to decide the allocation or bids in each round, respectively.

Appointment scheduling is an important research topic within the health care field. Gupta and Denton discuss the challenges and complexity of such a topic, reviewing appointment management systems identified in the literature [48]. The ability to respond in real-time is another important factor affecting the performance of the system. In light of this, Nan Liu et al. introduced a method for dynamic patient appointment scheduling. They developed a framework and heuristic dynamic polices that explicitly consider patient no-shows and cancellations, with the objective of assigning a proper date to each patient depending on the clinic's schedule at the time of request. They based their approach on an assumed positive relationship between *appointment delay* (defined as the gap between request date and actual appointment date) and cancellation/no-show rates. The proposed method can even take into account patient preferences, and uses simulation studies to show its effectiveness and efficiency, though it can only handle one request at a time. In the research of Ming et al., a simulation framework was developed for outpatient appointment issues, implemented as a decision support tool tackling real world problems [49]. The framework consists of four components: external demand for appointments, supply of provider time-slots, patient flow logic, and a scheduling algorithm.

Considerable effort has been put into rescheduling issues in manufacturing systems. According to the research of Vieira et al., rescheduling is not merely a collection of techniques, but also a control strategy, which has important impacts on system performance [50]. The authors performed a comprehensive review of manufacturing rescheduling procedures, and proposed a framework consisting of rescheduling strategies, policies and methods. They studied three rescheduling policies of predictive-reactive scheduling, namely periodic, event-driven, and hybrid. The periodic method is commonly used in situations where no online information is available during the current schedule period, and the dispatcher has to actively collect the information periodically. This approach yields schedule stability in dynamic manufacturing environments, but it is difficult to determine the optimal rescheduling period. Event-driven approaches are triggered by a single event, or when a number of events reach a certain threshold. Single event-driven methods require excessive computational resources when the event occurs rapidly, also requiring rapid data collection and processing capabilities. Various hybrid methods can be customized according to different specific requirements.

In [51], Weyns et al. introduced a DynCNET protocol, designed as an extension of a standard contract net, to dynamically assign tasks. It allows agents to dynamically change provisionally allocated tasks. The proposed protocol involves multiple participants who execute the tasks as well as multiple initiators who offer tasks. Interest area is introduced as a variable to initiate the scope of both participant and initiator agents. Both parties search for matches in their own particular area. The participant is allowed to switch to a new opportunity before the allocated task starts to execute. Similarly, the initiator is allowed to assign an allocated task to a new available participant before the task execution starts. A perception model is used to monitor both opportunities and availabilities. Protocol areas of interest are designed to reduce convergence risk, and the synchronization issue is handled by confirmation messages. The performance of DynCNET is evaluated by applying it to a transportation system.

Weiming and Norrie proposed a Mediator architecture, based on mediation mechanisms, as an agent-based solution for dynamic manufacturing scheduling and rescheduling [52]. It combines a Contract Net protocol with a bidding mechanism. The adapted protocol treats the un-awarded participants as alternative candidates for unforeseen situations in order to reduce rescheduling time. The alternatives are contacted directly when the awarded participant cannot perform the allocated tasks. Multiple mediators are involved hierarchically in the architecture, and the bidding, mediation and scheduling optimization are structured accordingly along different levels. The authors introduced several mechanisms to respond to different unexpected future events. The designed architecture is implemented on the MetaMorph II project they were currently working on.

Traditional negotiation protocols are based on full commitment contracts, in which the contracted parties must fulfill their contractual obligations. Correspondingly, Sandholm and Lesser designed a leveled-commitment contracting protocol as a backtracking instrument for multi-agent systems [53]. Differing from full-commitment contracts in traditional negotiation systems, leveled-commitment contracts are not bound to the same obligations; instead, they set de-committing penalties to allow agents to free themselves from the contract by paying the specified penalty to other contracted parties. According to their research, including de-commitment provisions in contracting can increase the expected payoff for all contracted parties. In many cases, de-committing a previous contract could represent a better option for practitioners in cases where there are no specific regulations or business processes that prohibit the breaking of the contract.

2.3 Other Resource Allocation Problems in Home Health Care

As previously mentioned, home health care scheduling is one important research direction of health care resource allocation problems. Other resource allocation problems in home health care settings, such as districting problems [4, 54, 55] and assignment problems [3, 56, 57], are also worth a brief review. Benzarli et al. focused their research on the districting problem concerning care delivery efficiency [4]. They built two mixed-integer programming models to balance the care workload. Based on some typical home health care cases in Europe, Lanzarone et al separated resource assignment aspects from rostering, routing and districting problems [3]. They built two models, based on deterministic patient demand and stochastic demand, respectively. The models share the same objective that balances practitioner workloads. Their study focused specially on the care continuity as well as consideration of skill and location constraints. Similarly, Errarhout et al. [58] developed a model based on generalized assignment problems, but which considered a greater number of constraints (e.g., caregiver qualifications and capacities) to better reflect real cases. In another work [6], assignment problems and routing problems were sequenced as a two-stage approach where the output of each assignment problem is incorporated as an input of the routing problem, with the authors focusing on the interaction between these two steps. Human dimensioning problems are another issue studied in home care operations management [59], [60], however any further discussion of such research is beyond the scope of this paper.

2.4 Summary

This chapter reviewed home health care resource scheduling problem models as well as existing solution approaches. We explained the rostering and routing characteristics of the model and categorized the solutions along centralized, decentralized, and dynamic perspectives. Among these three types of approaches, centralized approaches employ either exact, heuristic, or hybrid methods to solve scheduling problems, without considering the privacy of practitioners. Decentralized scheduling problems call for economic-based solutions where auction is the commonly used mechanism. Many research efforts have concentrated on traditional health care domains, while home health care scheduling problems are rarely studied, and even fewer concern themselves with agent intelligence with no effective scheduling algorithm. Other related decentralized scheduling problems employ contract net protocol as the interaction protocol for agents. In addition to traditional contracting protocols, leveled commitment contracting can be used when de-committing is allowed. Dynamic scheduling has been intensively studied in the manufacturing field, producing numerous quality solutions. Among the reviewed literature in the health care area, outpatient scheduling and appointment scheduling problems meet the requirement of handling unexpected events that demand a specific rescheduling approach. The selection of periodical, event-driven, and hybrid rescheduling are investigated from the trigger time dimension. Regeneration and repair of initial schedules need to be determined in the methodology dimension. Finally, the last part of this chapter briefly reviewed other home health care resource allocation problems.

The aim of this thesis is to develop a dynamic, decentralized scheduling approach to the home health care scheduling problem. In our chosen setting, the practitioners are self-interested in the economic setting, where their private information is unknown to others. As mentioned previously, auction is the commonly used mechanism in problems of market setting, and iterative auction based approaches do not require the agents to submit exact information about their private valuation. This approach also has the potential to support dynamic scheduling, which is the most important characteristic of our research topic. We expect that a dynamic, decentralized scheduling approach can be developed based on the careful investigation of the unique features of the home health care resource-scheduling problem.

Chapter 3 Home Visit Scheduling Problem

As previously mentioned, health care agencies are expected to dynamically allocate practitioners to home visits within the current operating schedule. In our setting, Home Visit Scheduling (HVS) is a decentralized problem, in the sense that the real cost of a practitioner on a home visit is private information, unknown to the health care agency. In addition, the dynamic nature of the home care environment means that schedulers cannot possibly know all the necessary information for generating a schedule until after that schedule takes effect, requiring updates to accommodate unexpected situations or changes. However, apart from the decentralized nature of the problem, we can safely assume that all required information is known at the moment of scheduling so long as we consider the problem as occurring in a short period of time and in a centralized environment. Thereafter, the problem becomes relatively static, with the number and availability of practitioners remaining unchanged and all home visits established at the time of scheduling. In this chapter, we first focus on the combinatorial optimization nature of this problem through a mathematical model. Following this, we discuss the decentralized and dynamic challenges represented by this problem.

3.1 Mathematical Model

Our primary goal is to find the optimal schedule from a set of feasible schedules. From the literature review we know that the conventional solution to the centralized scheduling problem is solving an optimization model through either exact, heuristic, or hybrid methods. This section focuses on the combinatorial optimization nature of this problem in a centralized environment. Future events are not our concern in a static environment, and all information regarding both visits and practitioners are known to the health care agency. The combinatorial optimization nature can be clearly demonstrated under such an environment in which we assume the agency

Symbol	Meaning
Ω	A set of visits
Π	A set of practitioners
X	A set of possible bundles of visit. Here the possible bundle is defined as the visit bundle with no interval overlap.
$T[\Omega][\Omega]$	T[i][j] expresses the travel time between visit i to visit j. Here we suppose the travel time from visit i to visit j equals the travel time from visit j to visit i.
V	An element in Ω
sp	Standard payment of a visit
x	The number of element in
m	The number of elements in Ω
n	The number of elements in
B	A bundle of visit
E	The union of the sets of feasible schedules from all practitioners
E_j	The set of feasible schedules of practitioner j
Υ	$\Upsilon[i]$ is an information set of practitioner $\Upsilon[0]$: The prefer departure time $\Upsilon[1]$: The prefer back time $\Upsilon[2]$: The hourly rate of work time $\Upsilon[3]$: The mileage cost per hour $\Upsilon[4]$: The preference violation cost of visit type 1 $\Upsilon[5]$: The preference violation cost of visit type 2 $\Upsilon[6]$: The preference violation cost of visit type 3 $\Upsilon[7..(7+m)]$: The travel time between home and visit[i-6]
$P[\Pi][\Upsilon]$	The personal information of each practitioner

Table 1 Symbols used in this thesis

knows the practitioners' costs. Based on this assumption, we can conveniently model the problem as a mixed integer program.

Consider an HVS problem consisting of a set of n practitioners, a set of m home visits, and a single health care agency. For a home visit i $(i = 1, \ldots, m)$, there is a standard payment sp_i, which is the upper limit that the agency would pay to cover the visit. It is assumed that the cost of covering visit i is less than sp_i for all practitioners. A home visit schedule defines a schedule of patient homes visited by practitioner. A practitioner j $(j = 1, \ldots, n)$ can configure his or her home visit schedule by selecting a bundle of visits. Let E_j be the set of home visit schedules which are feasible to practitioner j and E be the union of the sets of feasible schedules from all practitioners, $E = \bigcup_{j=1\ldots n} E_j$. Let $cost_j(B)$ be the cost to practitioner j required to complete the home visit schedule $B \in E$. $cost_j(B) < \sum_{i\in B} sp_i$ for all $B \in E_j$; $cost_j(B) = \infty$ for $B \notin E_j$. Let $x_j(B) = 1$ if the schedule $B \in E$ is allocated to practitioner j and zero otherwise. The HVS problem is about the selection of a set of home visit schedules for practitioners that covers all the planned home visits, while at the same time minimizing the sum of practitioner costs. The problem can be expressed using the following integer programming:

$$min \sum_{j=1}^{n} \sum_{B \in E} x_j(B) cost_j(B)$$

subject to

$$\sum_{B \in E} x_j(B) \leq 1\,, \quad j = 1, \ldots, n \tag{1}$$

$$\sum_{B \ni i} \sum_{j=1}^{n} x_j(B) = 1\,, \quad i = 1 \ldots m \tag{2}$$

$$\sum_{B \in E} x_j(B) = \sum_{B \in E_j} x_j(B)\,, \ j = 1, \ldots, n \tag{3}$$

$$x_j(B) = \{0,1\}, \ B \in E, \quad j = 1, \ldots, n \tag{4}$$

Constraint (1) ensures that a practitioner can only obtain one home visit schedule during the time window. Constraint (2) ensures that every planned visit is covered by a home visit schedule that has been allocated to a practitioner. The set of constraints (3) ensures that if a schedule is assigned to a practitioner then it must belong to the whole set of feasible schedules of the practitioner. Together, these constraints prevent the agency from assigning practitioners schedules that they are unable to take. Constraint (4) is a set of integer constraints. The HVS problem is NP-hard, as stated in the following theorem.

Theorem 1: The problem of Home Visit Scheduling (HVS) is NP-hard.

Proof: To show that HVS is NP-hard, consider a special case in which $E_j = E$ for all $j = 1, \ldots n$. In this case, Constraints (3) always hold. The relaxed model is a set covering problem, which is NP-complete. It follows that, as a general case, HVS problem is NP-hard.

Constraints (2) ensure that every planned visit is covered by a home visit schedule which has been allocated to a practitioner. To obtain a feasible solution to the centralized model, constraints (3) must hold. To ensure that constraints (3) always hold, we assume that there exists a subset of practitioners, call them backup practitioners, for each of them, their feasible schedule set only contains one schedule and that schedule only covers one visit. We also assume that together the group of backup practitioners will cover all planned visits.

3.2 The Challenge of Decentralized Environments

This model has been built without considering the distributed characteristics of the HVS problem. However, the practitioners in our setting are independent health care service providers, not employees of the health care agency, and as such their relationship with the agency is a type of contractual relationship. In such a setting, scheduling-related information is distributed across all involved parties. The health care agency lacks a global view of the problem, as does any one of the practitioners, since information like the cost of each individual practitioner's visits are known only by the practitioners themselves. It is therefore necessary to construct a decentralized environment, in which the practitioners are assumed as intelligent, rational, and self-interested agents with sufficient autonomy to control their own schedules independently. Similarly, the health care agency is the other agent in the model, making its decisions in the same way.

In the centralized model, we can easily find the optimal solution, since all information is known. Following the same line of thought, the agency in the described environment is also able to find the optimal solution by centralizing all the needed information from the practitioners. However, the optimality of the schedule depends on the accountability of the reported information. In our thesis, practitioners are treated as self-interested agents in the sense that they maximize their own payoff without considering others' interests, with no control for the practitioners to report true information. Thus, we need a mechanism capable of motivating the practitioners to report truthful information, by which their objectives are maximized only through reporting truthful information.

In the HVS problem of such an environment, the agents take action to maximize their own objectives, having neither control over each other's actions nor any interest in the other agents' objectives. However, they are allowed to cooperate in order to achieve their respective goals in case the cooperation benefits them. A negotiation mechanism that allows coordination, while at the same time creating incentives for the practitioner to tell the truth would satisfy our requirements. From our review we found that economic-based models address such incentive issues, while auctions, as an application of the economic mechanism, are useful in solving decentralized scheduling problems. Based on insights into the combinatorial nature of HVS, it is natural to employ combinatorial auctions to solve the problem. In an HVS auction, the practitioners compete to obtain a bundle of visits from the health care agency, reversing the roles of buyer and seller. The next chapter describes how we embed the designed reverse combinatorial bidding framework in an agent-based scheduling system for home health care.

3.3 The Challenge of Dynamic Environments

The dynamic characteristics of the home health care landscape further increase the complexity of HVS. It is highly possible for there to be a new visit request, cancellation, or changes in practitioner availability during the time horizon of the practitioners' schedules. The HVS problem requires us to accommodate such unexpected events during the schedule's operation and to continually revise schedules in a cost-effective manner.

The most common scheduling changes in home health care are new incoming visits, cancellations, visit time changes, and practitioner availability changes. We can group these situations into two categories, based on whether they release a practitioner from an assigned visit or assign a practitioner to a previously unassigned visit. While the first and second events mentioned above are straightforward, it is helpful to provide examples to better explain the event handling process in the other cases. The first scenario involves a patient calling to change their visit time. Such an event can be separated into two steps: cancelling an allocated visit, and assigning a practitioner to an incoming visit with an updated visit time. Another scenario is that the practitioner's availability changes during the current schedule. In some cases this makes the visit impossible and requires a cancellation. The cancelled visit is taken as an incoming visit, which should then be allocated to another practitioner. In other cases cancellations may result in a practitioner having a newly available timeslot that can then be allocated to a new visit request. The third example is that the service start time is considered as a hard constraint that cannot be violated. Under such an assumption, the visit delay leads to a visit cancellation. This thesis chooses to focus on the most challenging situation, that of accommodating the incoming visit requests. As we described previously, the HVS problem is considered as a set of static scheduling problems, while the input is the current state of the system. Solving such a static problem requires us to reschedule the plans every time there is a change, based on the latest information, which should be collected at the moment of rescheduling. As such, *when* we collect information and conduct rescheduling is the first question that needs to be answered.

In terms of this question, periodic rescheduling and event-driven rescheduling are studied in the literature. Periodic rescheduling means the plan is supposed to be rescheduled after a predefined time interval. The second approach means that the system conducts rescheduling each time a schedule-related event occurs. The hybrid method, which combines periodic and event-driven rescheduling, is yet another method where the rescheduling is triggered when an event occurs and at the end of a time interval.

For a periodic rescheduling strategy, the rescheduling interval needs to be investigated depending on specific problems. The incoming visit has to be handled in a given amount of time before the visit commences in order to prepare the necessary paperwork. However, such events occur stochastically in the HVS problem, and the time between the patient calling and the newly scheduled visit is uncertain. It is therefore difficult to determine the proper rescheduling interval for such situations. Normally, the home health care agency does not handle urgent medical demands in the real world. Last minute changes or urgent cases are considered exceptions within the HVS problem, as we suppose that patients typically make appointments with health care agencies for home visits. There are two reasons we did not select a pure event-driven method for the HVS problem. Firstly, it consumes unnecessary computing resources in cases where the changes occur rapidly, while the health care agency does not necessarily need to handle the event immediately. The goal of our system is searching for an optimal or near-optimal solution rather than merely responding quickly. Secondly, better solutions become possible only if the health care agency considers multiple events together. Contrastingly, in cases where the event needs to be handled immediately, the primary goal of the system is to make the necessary changes promptly, rather than taking the additional time required to find the most optimal solution. As such, we can safely conclude that hybrid rescheduling is a better choice for solving the HVS problem.

The other question that needs to be answered is *how* to conduct the rescheduling. In general, there are two main rescheduling strategies: regeneration and repair. The first strategy is to completely regenerate the schedule for all remaining visits. While this is a feasible method in principle, changing practitioners' schedules without their permission is likely to lower their satisfaction and produce a more chaotic work environment. Another practical issue is that there is typically paperwork that must be completed at least one day prior to such changes. Sometimes this involves coordinating with an agency, collecting medial materials, or going through an approval process. As such, the repair strategy appears to be a more appropriate method for the HVS problem. We employ a repair-based approach to allocate incoming visit requests, at the same time keeping the already allocated visits unchanged. In this way, we minimize the changes to the original schedule for practitioners.

In summary, HVS is expected to maintain a dynamic schedule for practitioners through its handling of unexpected events. We focus on accommodating incoming visits in particular. In this system, existing visits in practitioners' schedules remain unchanged, while new visit requests are allocated. Schedule repair can be performed in either a flexible time interval or triggered by events depending on specific situations.

Chapter 4 Decentralized Scheduling for HVS

The objective of home health care resource scheduling is to reduce the cost of providing care through effective, efficient practitioner allocation. Since the problem must be solved in a decentralized environment, an agent-based scheduling system is expected to best construct schedules while solving the problem. In this chapter, we first introduce the design of the multi-agent system, followed by the local decision-making process for each agent. An iterative bidding mechanism is proposed as the negotiation protocol for the scheduling system. Following this, a worked example is presented to demonstrate exactly how the system works. Finally, we conduct a computational study in order to evaluate the proposed solution's performance.

4.1 Multi-Agent System Design

The design of a multi-agent system for decentralized home visit scheduling consists of the architecture, the negotiation protocol, and the system initialization. Two types of agents are recognized, and it is assumed that they work collaboratively to achieve their goals. The proposed protocol is used in the coordination process between agents. An initialization process before the start of negotiation is described as well.

4.1.1 System Architecture

As shown in Fig. 3, there are two distinct types of agents in this architecture: Practitioner Agents and the Health Care Agency. The Practitioner Agent functions as a personal assistant to practitioners, maintaining their schedules, while the Health Care Agency (HC Agency) represents the home visit scheduler. Based on the architecture, the HC Agency collects home

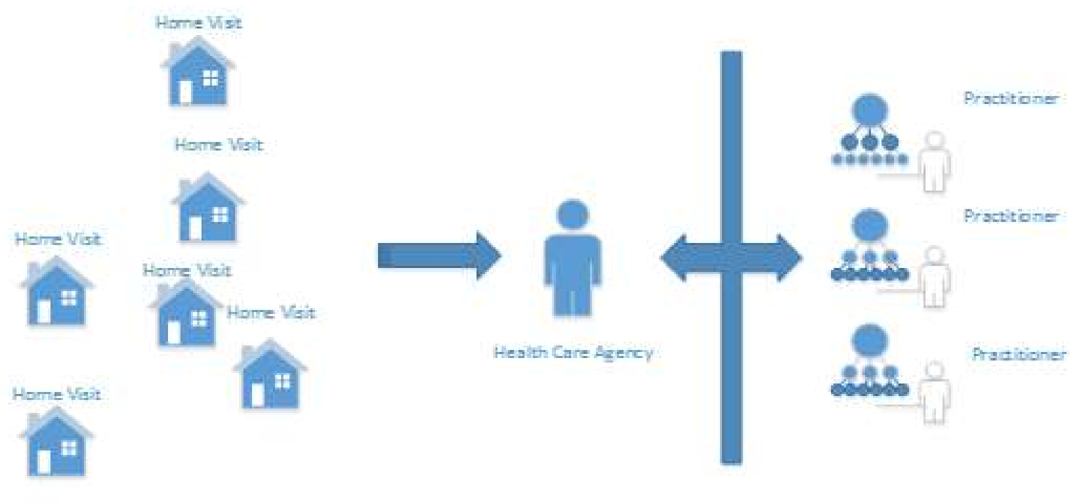

Figure 3 System Architecture

visit requests and facilitates negotiation between agents to construct the practitioners' schedules. During the negotiation process, both Practitioner Agents and the HC Agency make their decisions based on their individual objectives, as well as other agents' responses.

4.1.2 Negotiation Protocol

The scheduling process can be seen as a coordination procedure between the HC Agent and a group of Practitioner Agents, during which a negotiation protocol is used to allocate incoming visit requests to practitioners. In this system, we propose an iterative bidding mechanism as the negotiation protocol. This is a price mechanism, in which the health care agency coordinates the procurement of medical services from independent practitioners by adjusting the prices of home visit schedules. The negotiation protocol contains the following four steps:

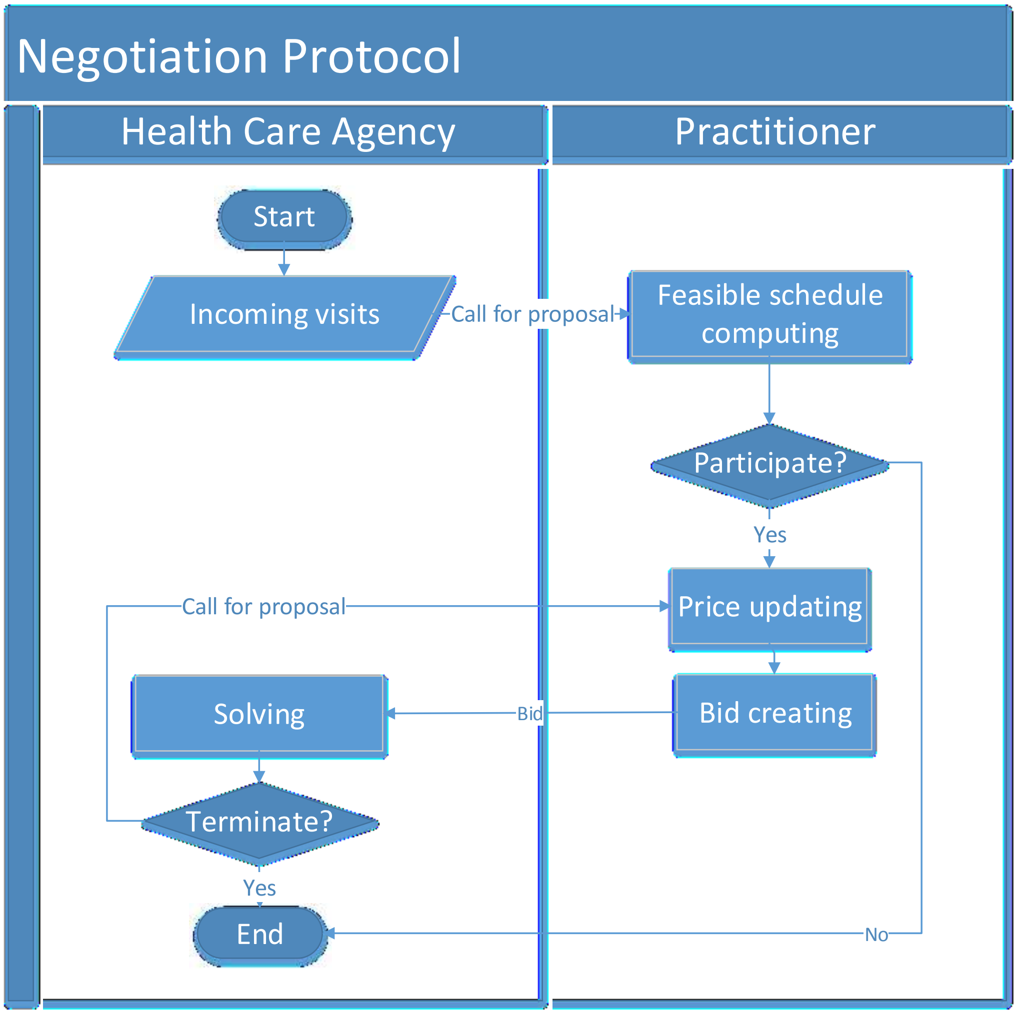

Figure 4 Negotiation Protocol

1) The HC Agency recognizes the time of scheduling. The HC Agency firstly collects information on the set of visits waiting for allocation. It then solicits bids by sending a call for proposal message containing the visit information to registered Practitioner Agents.
2) Next, each Practitioner Agent decides for his or herself whether or not to participate. Willing agents send a bid back to the HC Agency, including the schedule, charge, and the Practitioner Agent's identification.
3) Upon receiving bids from all Practitioner Agents, the HC Agent computes the visit allocation, taking all the bids as input and decides whether to accept or refuse each bid. After the decision is made, the HC Agency notifies all participating Practitioner Agents with the results.

4) The negotiation iteratively goes through the above two steps, for which a reverse combinatorial bidding framework is implemented in the designed system. The Practitioner Agents update prices and bid again depending on the results of the previous bidding round. Upon receiving bids, the HC Agency decides the final round if the termination condition is satisfied and sends the final results to the Practitioner Agents. Alternatively the HC Agency may decide to continue the process by sending another call for proposals to all participating Practitioner Agents.

4.1.3 System Initialization

Some information used throughout the negotiation process will be initialized at the beginning, such as the set of visit requests, the payment of each visit, and the feasible packages of each Practitioner Agent. In this stage, the practitioners first calculate the potential payoffs of their feasible schedules and then bid on the schedule with the highest payoff (breaking ties randomly). The details of the initialization process are described as follows.

At the beginning, the health care agency sends a call for proposal message to the practitioners with details on the set of planned visits and the upper-limits of the service charges that they can pay for each of the visits, which is referred to as the standard price for the visit. Practitioners compute their respective sets of feasible schedules E_j. For each schedule E_j, the practitioner computes their cost of serving it. The standard price of a schedule is the summation of the standard prices of visits covered by the schedule. The initial bidding price for a schedule is set to be equal to its standard price. With known costs and initial bidding prices of schedules in their E_j, a practitioner computes the payoff of taking each schedule. Considering the private value model mentioned above, a practitioner's payoff for a schedule is the remainder of the payment after deducting their costs from the bidding price at each round of bidding. The practitioner must be compensated by their cost of servicing the schedule to keep a positive payoff. After initialization, the system goes through an iterative bidding process.

4.2 Practitioner Agent's Local Decision Making

After receiving the call for proposal from the HC Agent, Practitioner Agents compute their feasible packages, calculate the payoffs, and create bids for the highest utility package (breaking ties randomly) at each round. We use a schedule-charge pair ⟨schedule, charge⟩ to represent the practitioner's bid, where schedule is the set of visits that the practitioner wants and charge is the compensation that the practitioner requires for providing the services, which is bidding price in this framework. The service charge structure is practitioner-dependent and non-anonymous, meaning there is no common service charge for a schedule. This structure allows the practitioners to price the same schedule differently according to their own cost structure and time preferences.

At each round t ($t > 1$), practitioners start by updating their bidding prices for the schedule submitted at round $t-1$. There are three different scenarios for practitioners to act out at round t, depending on the provisional allocation status determined in round $t-1$: (1) if a practitioner's bid was not awarded in the provisional allocation at round $t-1$, he/she can decrease the bidding prices by ε on the schedule bid for in round $t-1$ or rounds before $t-1$, where ε is the minimum price decrement imposed by the health care agency. Since practitioners are assumed to be rational profit maximizers, they generally do not bid with a decrement more than ε; (2) another choice for practitioners who are not assigned a schedule in round $t-1$ is keeping the same bidding prices (add back the ε amount). However, if a practitioner adds this ε back to the bidding price, they will be considered to have entered their final bid status and they are not allowed to decrease the bidding prices on any of their schedules in future rounds. It may happen when the payoffs of every other schedule become negative; and (3) the practitioner can also withdraw from bidding. If a practitioner is provisionally assigned a schedule in round t-1, they may want to keep their bidding price unchanged in the next round, which means they are allowed to repeat the same bids in round t. However, practitioners are not prevented from entering a lower bid in future rounds in this scenario.

After updating the bidding prices, a practitioner needs to compute the payoff of each schedule again based on the updated bidding prices to determine the maximum payoff schedule. In computing such a schedule or schedules, a practitioner j solves a maximization problem

$max_{B \in E_j}[p_j^t(B) - cost_j(B)]$ and obtains a set of schedules with an equally maximized payoff, where $p_j^t(B)$ represent the bidding price of schedule B at round t. That is, for any two schedules B and B' in the payoff maximizing set, $p_j^t(B) - cost_j(B)$=$p_j^t(B') - cost_j(B')$. After this, the practitioner randomly chooses one from the set of schedules with a maximum payoff, and bids for that schedule with the updated bidding price. In situations where a practitioner has entered into final bid status, he/she is no longer allowed to decrease the bidding price. However, the practitioner can repeat their final bid in future rounds until termination. We set up this final bid repeating arrangement to allow the temporarily excluded bids to come back to the game to further reduce the health care agency's overall costs. In the iterative bidding process, some bids can be temporarily "excluded" from the provisional allocation because there is a particular combination of allocation constraints and resource requirements with lower overall service charge in each specific round. Along with ongoing bidding, this particular situation may have changed to allow the previously excluded bids back into the bidding process. However, those bids will not be submitted again without this setting if their costs have been reached during the "excluded" periods, which means the practitioner will not choose to bid them, even though some visits in them become available in subsequent provisional allocations.

4.3 Health Care Agency Local Decision Making

In the HC Agency's decision-making process, we assume there are always backup practitioners willing to take visits at a standard price, meaning the HC agency can assign a backup practitioner to the visit at any time as needed. At the local level, the HC Agent solves a winner determination problem to allocate practitioners to visits. It first screens out the invalid bids from all bids received from practitioners. Those bids will not enter into the winner determination procedure. There are two types of invalid bids: (1) any bids with bidding prices above the lowest one for that same schedule received in previous rounds, (2) bids with decreased prices from practitioners who have already declared their final bidding status previously.

The health care agency then checks the termination condition against the valid bids. The bidding will terminate at the round with no price updates for all valid bids, which means all practitioners participating in bidding in this round have repeated their bids. After the bidding terminates, the health care agency allocates schedules to practitioners according to the final allocation at their bidding prices. If the termination condition is not satisfied and the procedure continues, the winner determination model will take the set of valid bids as input and solve the problem. The auction goes back to price updates and bidding after the winner determination.

The winner determination will compute the provisional allocation each time there is new input from the last stage, which means the termination checking fails in this round. In order to cover all visits in the provisional allocation, while at the same time minimizing the overall price, the winner determination is modeled to select a subset of the bids submitted by practitioners which satisfy the covering and minimizing constrains. Let N^t be the set of practitioners submitting their bids at round t and $p(B_j^t)$ be the bidding price of B_j^t, where B_j^t is the schedule submitted by practitioner j at round t, $j \in N^t$. Let $Z_j = 1$ if practitioner j wins and $Z_j = 0$ otherwise. The winner determination can be modeled to an integer programming as following.

$$min \sum_{j \in N^t} Z_j p(B_j^t)$$

subject to

$$\sum_{j \in N^t \, B_j^t \ni i} Z_j = 1, i = 1 \dots m \quad (5)$$

$$Z_j = \{0,1\}, \quad j \in N^t \quad (6)$$

Constraints (5) ensure that the bids awarded in a provisional allocation will cover all the planned visit requests. Constraint (6) is a set of integer constraints.

4.4 A Worked Example

In this section, a worked example is presented to demonstrate the designed system. Suppose a health care agency allows its contractual practitioners to customize their schedules. The visit information is available for both the health care agency and practitioners. Given the visits, each practitioner will assemble their feasible schedules by calculating both utility and time constraints for a specific time window. A feasible schedule generation method is introduced as following.

- **Feasible Schedule Generation**

Within the proposed iterative bidding framework, practitioners are expected to make multiple decisions during the bidding process. They need to find feasible schedules, calculate the utility for each, and update their bid price in each round. The utility calculation and price updating rules have already been introduced in the previous section. In this section, we demonstrate a method to partially aid practitioners in generating feasible schedules. The feasible schedule of a specific practitioner is defined as a schedule that: 1) satisfies all hard and soft constraints; 2) has no time conflicts between the feasible schedule and the current schedule; and 3) should have positive utility for the practitioner it belongs to.

To model this problem, we first introduce a schedule cost structure. The cost of a given schedule in the proposed problem consists of three parts: time, transportation, and preferences. The time a schedule practitioner spends on each visit includes visit time, travel time, idle time between each visit in the schedule, and paperwork time. The paperwork time will not be explicitly addressed because it is considered an irrelevant constant that will not affect the effectiveness of the calculation. Besides this, practitioners will have transportation costs incurred as they travel from one visit to another. The self-interest of the practitioners leads us to assume they prefer some visits to others. This means there will be a preference violation cost reflecting the level of the preference violation to the practitioner. The feasible schedule generation process contains two steps: 1) find out all possible schedules, and 2) estimate the cost of each and determine feasible schedules.

Suppose the feasible schedule generation consists of a given set of m medical visits V, and each visit has a standard price sp_i, a service start time $s_{i,}$ and a service end time e_i. We assume the travel time from visit A to visit B is same as that from visit B to visit A. The modality of transportation is not concerned. In order to find the feasible schedules E_j for a practitioner j, we firstly find all subsets of V and remove the subsets with visit time overlap. We then sequence the visits in each schedule by start time s_i and then by end time e_i. After that, we add the travel time from v_i to v_{i+1} to the end time of v_i , denoted by e'_i . The schedules with visit time overlap considering travel time need to be removed from the subsets. As a consequence, we have a possible schedule set for all practitioners, denoted by X. For a specific practitioner, a feasible schedule would be that which accommodates their individual schedule and at the same time has

Visit ID	Start Time	End Time	Type	Standard Price
1	12:00	13:00	1	$90
2	13:00	14:00	1	$90
3	14:00	15:00	2	$30
4	15:00	16:00	2	$30
5	16:00	17:00	2	$30
6	17:00	18:00	2	$30
7	11:00	12:00	2	$30
8	12:00	13:00	2	$30
9	13:00	14:00	2	$30
10	14:00	15:00	2	$30
11	15:00	16:00	2	$30
12	16:00	17:00	2	$30
13	13:00	14:00	3	$60
14	14:00	15:00	3	$60
15	15:00	16:00	3	$60

Table 2 Summary of Visits

positive utility. We assume that practitioners depart from their homes and end their day once they return home. The practitioners' own schedule concerns the departure time of the first visit and the end time of the last visit once they have finished the whole schedule. Given the travel time between visits and the time value between the practitioner's home j and the visits, we know the travel time of the schedule $B_m \in E_j$. The transportation cost M_{jm} is calculated by the travel time of B_m multiplied by a mileage rate of practitioner j. The work time is the duration from the practitioner's departure to their return back home, including visit time, travel time, and idle time. Thereafter, the work-time cost T_{jm} spent on a schedule B_m for practitioner j will be the work time multiplied by the hourly rate of work. Additionally, the practitioner can setup a preference violation cost for each type of visit contained in a schedule, which sum to a total preference violation cost O_{jm}. It is essentially a subjective money value adopted by an agent for each type of visit. For example, a practitioner would rather get $10 less payment for a re-visit over a new admission, because the practitioner is familiar with the re-visit patient and such a visit is much easier. The cost of schedule $B_m \in E_j$ for practitioner j would be the sum of those three costs: $T_{jm} + M_{jm} + O_{jm}$.

0	0.27	0.38	0.43	0.52	0.33	0.45	0.67	0.68	0.65	0.32	0.57	0.5	0.48	0.62
0.27	0	0.7	0.3	0.6	0.3	0.6	0.55	0.67	0.65	0.25	0.35	0.58	0.65	0.3
0.38	0.7	0	0.55	0.25	0.48	0.73	0.45	0.63	0.47	0.47	0.55	0.53	0.7	0.65
0.43	0.3	0.55	0	0.4	0.68	0.55	0.4	0.73	0.62	0.5	0.37	0.7	0.32	0.3
0.52	0.6	0.25	0.4	0	0.27	0.27	0.4	0.6	0.48	0.38	0.73	0.37	0.55	0.42
0.33	0.3	0.48	0.68	0.27	0	0.65	0.62	0.67	0.47	0.58	0.47	0.33	0.42	0.32
0.45	0.6	0.73	0.55	0.27	0.65	0	0.63	0.5	0.58	0.67	0.27	0.72	0.38	0.43
0.67	0.55	0.45	0.4	0.4	0.62	0.63	0	0.47	0.5	0.72	0.6	0.4	0.42	0.68
0.68	0.67	0.63	0.73	0.6	0.67	0.5	0.47	0	0.37	0.55	0.5	0.38	0.53	0.48
0.65	0.65	0.47	0.62	0.48	0.47	0.58	0.5	0.37	0	0.72	0.27	0.42	0.32	0.45
0.32	0.25	0.47	0.5	0.38	0.58	0.67	0.72	0.55	0.72	0	0.28	0.5	0.57	0.43
0.57	0.35	0.55	0.37	0.73	0.47	0.27	0.6	0.5	0.27	0.28	0	0.53	0.72	0.47
0.5	0.58	0.53	0.7	0.37	0.33	0.72	0.4	0.38	0.42	0.5	0.53	0	0.73	0.28
0.48	0.65	0.7	0.32	0.55	0.42	0.38	0.42	0.53	0.32	0.57	0.72	0.73	0	0.67
0.62	0.3	0.65	0.3	0.42	0.32	0.43	0.68	0.48	0.45	0.43	0.47	0.28	0.67	0

Table 3 Travel Time between Visits (hours)

Practitioner	Feasible Schedules		Cost	Payment
P1	B(1,1):	<7,2,4>	$149.82	$150.00
	B(1,2):	<1,14,5>	$169.35	$180.00
	B(1,3):	<1,14,12>	$169.17	$180.00
	B(1,4):	<2,15,6>	$167.16	$180.00
	B(1,5):	<7,2,15>	$157.87	$180.00
P2	B(2,1):	<1,14,5>	$176.12	$180.00
	B(2,2):	<2,15,6>	$178.83	$180.00
	B(2,3):	<7,2,15>	$177.21	$180.00
P3	B(3,1):	<7,2,4>	$125.83	$150.00
	B(3,2):	<7,2,5>	$138.43	$150.00
	B(3,3):	<7,2,11>	$124.90	$150.00
	B(3,4):	<7,2,12>	$143.29	$150.00
	B(3,5):	<7,2,15>	$128.08	$180.00
	B(3,6):	<13,15,6>	$147.42	$150.00
	B(3,7):	<7,13,15>	$137.11	$150.00
P4	B(4,1):	<1,3,5>	$123.90	$150.00
	B(4,2):	<1,10,5>	$124.10	$150.00
	B(4,3):	<1,3,12>	$129.12	$150.00
	B(4,4):	<1,14,5>	$129.06	$180.00
	B(4,5):	<1,14,12>	$134.23	$180.00
P5	B(5,1):	<1,3,12>	$148.97	$150.00
	B(5,2):	<1,10,12>	$148.96	$150.00
	B(5,3):	<1,14,5>	$155.97	$180.00
	B(5,5):	<1,14,12>	$154.13	$180.00
	B(5,6):	<7,2,15>	$149.41	$180.00

Table 4 Practitioners' Feasible Packages and Corresponding Cost and Payment

- **Worked Example**

The visit information consists of two parts. The properties of an individual visit include start time, end time, type of visit, and the standard price (Table 2), as well as the travel time between visits, which in this example is randomly generated in hours (Table 3). On the other hand, the practitioners have their own preferences with respect to work time and the type of visit (Table 4). The travel time between practitioners' homes to each visit are also randomly generated. We intentionally keep the example oversimplified for the purpose of clearly illustrating the steps of the bidding process. The feasible schedules and their costs for practitioners are generated using the method described in the section "Feasible Schedule Generation". Table 5 lists each practitioner and his/her feasible schedules, the costs involved, and the payment. B (a, b) represents the feasible schedule b from practitioner a. As defined previously, there are backup practitioners B6 to B20 with each having only one feasible schedule containing one of the 15 visits respectively. These schedules are represented by B(6,1) to B(20,1), none of which are shown in the table. The backup practitioner will bid for the schedule at the standard payment at each round of bidding. To limit the number of rounds of bidding, the price decrement ε is set to \$15. Submitted bids, provisional allocation, agency's payment and practitioner's cost at each round of bidding are summarized in Table 6.

The auction terminates at round 11 with an overall solution cost of \$557.14. Compared with the optimal overall cost of \$557.14, computed using the centralized model, the auction reaches 100% efficiency in this example. The sum of the payments paid by the agency is \$615, which is close to the overall solution cost. The provisional allocations along the bidding process manifests the heuristic search path guided by the bidding mechanism.

Property	P1	P2	P3	P4	P5
Departure time	9:00	8:00	8:00	11:00	10:00
Back time	18:00	20:00	18:00	18:00	18:00
Hourly rate	$25	$28	$18	$20	$21
Mileage cost	$0.4	$0.3	$0.3	$0.4	$0.6
PVC of typ1	$1	$1	$4	$1	$3
PVC of typ2	$5	$5	$8	$5	$8
PVC of typ3	$10	$10	$13	$10	$13
Time to visit1	0.47	0.42	3.72	0.27	0.6
Time to visit2	0.45	0.28	0.53	0.72	0.23
Time to visit3	0.58	0.42	0.6	0.55	0.57
Time to visit4	0.3	0.73	0.55	0.73	0.68
Time to visit5	0.63	0.28	0.25	0.35	0.62
Time to visit6	0.57	0.52	0.55	0.47	0.32
Time to visit7	0.23	0.37	0.3	0.43	0.5
Time to visit8	0.7	0.25	0.58	0.7	0.27
Time to visit9	0.45	0.58	0.27	0.58	0.48
Time to visit10	0.33	0.25	0.43	0.4	0.6
Time to visit11	0.58	0.25	0.5	0.42	0.72
Time to visit12	0.62	0.58	0.52	0.6	0.53
Time to visit13	0.55	0.73	0.72	0.37	0.58
Time to visit14	0.47	0.45	0.53	0.45	0.65
Time to visit15	0.42	0.37	0.4	0.38	0.42

Table 5 Practitioner Information

Round #	Submitted bids	Provisional allocation	Payment	Cost
1	B(1,5),B(2,1),B(3,5),B(4,4),B(5,5),B(6..20,1)	B(1,5),B(2,1),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$660.00	$633.99
2	B(1,5),B(2,1),B(3,5),B(4,4),B(5,4),B(6..20,1)	B(3,5),B(6,1),B(8,1),B(9,1),B(10,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1), B(18,1),B(19,1)	$645.00	$608.08
3	B(1,4),B(2,3),B(3,5),B(4,4),B(5,3),B(6..20,1)	B(3,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$630.00	$557.14
4	B(1,3),B(2,2),B(3,5),B(4,4),B(5,5),B(6..20,1)	B(3,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$630.00	$557.14
5	B(1,2),B(2,2),B(3,5),B(4,4),B(5,4),B(6..20,1)	B(3,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$630.00	$557.14
6	B(1,5),B(2,2),B(3,5),B(4,4),B(5,3),B(6..20,1)	B(1,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$630.00	$586.93
7	B(1,5),B(2,2),B(3,3),B(4,4),B(5,2),B(6..20,1)	B(1,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$630.00	$586.93
8	B(1,5),B(2,2),B(3,1),B(4,4),B(5,1),B(6..20,1)	B(1,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$630.00	$586.93
9	B(1,5),B(2,2),B(3,5),B(4,4),B(5,5),B(6..20,1)	B(3,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$615.00	$557.14
10	B(1,1),B(2,2),B(3,5),B(4,4),B(5,5),B(6..20,1)	B(3,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$615.00	$557.14
11	B(1,1),B(2,2),B(3,5),B(4,4),B(5,5),B(6..20,1)	B(3,5),B(4,4),B(8,1),B(9,1),B(11,1),B(13,1),B(14,1),B(15,1),B(16,1),B(17,1),B(18,1)	$615.00	$557.14

Table 6 Submitted Bids, Provisional Allocation, Payment and Cost at Each Round of Bidding

4.5 Computational Study

In this section, we evaluate the performance of the bidding framework by comparing it with a VCG auction through a computational study. According to the experiment results, this framework reduces the computation required of the health care agency and partially reveals the private information of practitioner agents. However, these benefits are generally obtained with a cost to efficiency.

4.5.1 Design of Testing Data

In the design of the testing data, the standard payment for a schedule is the sum of the standard payments of visits included in the schedule. Practitioners will initially bid from the standard payment to maximize their payoffs. The data needed for cost estimation, such as the start and end time of visits, the basic information of each practitioner, and the travel time information, are all generated randomly in different ranges. Practitioners' costs for a particular schedule are calculated according to the description of "feasible package generation". We assume the health care agency plans the schedule in a relatively small time window, and we keep the number of planned visits in proportion to the number of practitioners in each problem instance. Ten instances are randomly generated, with 15 visits and 5 practitioners. In large scale data testing, the maximum number of visits included in one schedule should be considered, since, given a time window, a practitioner can only take on a certain number of visits. For example, the maximum number of visits taken by one practitioner is limited to eight, assuming a one-day time window. Backup practitioners can only take on a single visit in the experiment. The following two algorithms describe the process of feasible schedule generation.

- **Schedule Generation**

```
Algorithm 1 generateSchedule
Input: Ω, T
Output: X
  m1 ← m ← |Ω|
  Step1: Generate a list ‘Ω, where ‘Ω[i] ← V.vid (0 ≤ i ≤ m − 1)   (V ∈ Ω)
  Step2: Generate the non-empty power set of ‘Ω and transfer into a list Ω’
  m2 ← m3 ← |Ω’[m1]| (0 < m1 ≤ 2^m − 1)
  Step3: Generate the list of visit bundles X
  for 0 < m1 ≤ 2^m − 1do
```

```
    Step4: Generate the visit bundle X from the power set of 'Ω
    Step5: Sort X by X[m1][m2].s and X[m1][m2].e
    X[m1][m2].e' ← T[m2][m2 + 1] + V[m2].e
    Step6: if X[m1][m2].e' > X[m1][m2 + 1].s do
              remove X[m1] from X
           end if
  end for
```

Table 7 Schedule Generation Algorithm

This algorithm generates feasible schedules for each practitioner. The first input **Ω** is a set of visit V<vid, s, e, t, sp, a>, where vid is the identification of the visit. The identifications are consecutive integer numbers starting from 1; s: the start time of the visit; e: the end time of the visit; t: the type of the visit, $t \in \{1,2,3\}$ sp: the payment for the individual visit; and a indicates whether the visit is assigned or not in each round of bidding. The second input T is a $m \times m$ matrix. $T[i][j]$ is the travel time between visit i to visit j. Here we suppose the travel time from visit i to visit j equals the travel time from visit j to visit i. The output: X is a list of bundles, which represent all the possible schedules with no time overlap considering the travel time.

- **Cost Estimation**

```
Algorithm 2 estimateCost
Input: generateSchedule(), P
Output: E_j
for 0 < j ≤ n do
  scheduleID ← 1
  for 0 < j ≤ x do
    Worktime ← null
    Traveltime ← null
    PVC ← null
    f ← the index of the first visit of X[i] in P[j]
    l ← the index of the last visit of X[i] in P[j]
    q ← the index of preference violation cost of visit V in P[j]
    RequiredDepartureTime  ← First(X[i]).s-P[j][f]
    RequiredBackTime  ← Last(X[i]).e+P[j][l]
    Step3: Generate feasible schedule
    if P[j][0] ≤ RequiredDepartureTime and P[j][1] ≥ RequiredBackTime do
       Step3.1: Calculate the payment of schedule
       Step3.2: Estimate the cost of the schedule
       NetWorkTime ← Last(X[i]).e-First(X[i]).s
       HomeToWork ← P[j][f]+P[j][l]
       WorkTime ← NetWorkTime + HomeToWork
       TravelTime ← HomeToWork + BetweenCities (0 if only one visit);
       for all V in X[i] do
         PVC += P[j][q]
       end for
       CostOfSchedule[j][i] = WorkTime * P[j][2] + TravelTime * P[j][3] + PVC
       Step3.3: Generate the feasible schedule E_j
       scheduleID ++;
```

```
      end if
   end for
end for
```

Table 8 Cost Estimation Algorithm

The input P is a $n \times b$ matrix. P[j][k] is the value of the k^{th} property in ϒ for practitioner j. The output E_j is a set of feasible schedules <fid, D, c, p, z> for practitioner j, where fid is the identification of the schedule. The identifications are consecutive integer numbers starting from 1; D is a list of visit IDs representing the visits contained in the referred feasible schedule of practitioner j; c is the cost of the schedule; p is the payment of the schedule; and z indicates whether the practitioner bid this schedule or not in each round of bidding. P[j][0] and P[j][1] are the preferred departure and return time of practitioner j, respectively.

4.5.2 Experimental Results

The efficiency of the iterative bidding framework is evaluated in terms of the difference between the overall cost of the solution generated by the proposed framework and the optimal solution cost computed from the centralized model. For a problem instance, the optimal solution cost is computed by solving the centralized integer-programming model. The model is coded in ILOG OPL Optimization Programming Languages, (http://www-01.ibm.com/software/commerce/optimization/modeling/) and solved using ILOG CPLEX.

Comparison of the solutions computed by the iterative bidding model and the optimal ones computed by ILOG CPLEX is illustrated in Table 9. The fourth column and the fifth column show the optimal solution costs and the costs computed by the iterative bidding across all ten instances. All customers are assumed to adopt final-bid-repeating and ε is \$15 for all groups. It is observed that the cost of solutions generated by the iterative bidding framework is on average 4% higher than the optimal cost across the ten groups of problem instances. If we consider the optimal cost as 100% efficiency, the proposed iterative bidding framework achieves 98% of the efficiency of the optimal solution. It is also observed that the agency pays the practitioners on average 11% over their costs of taking a visit.

Epsilon = 15 Visit Number = 15 Practitioner Number = 5

Group	Optimal solution cost	Bidding solution cost	Bidding solution Payment	Initial bidding price
1	$708.738	$723.049	$750	$900
2	$671.121	$677.864	$780	$930
3	$764.919	$788.43	$825	$990
4	$882.902	$888.859	$975	$1050
5	$571.574	$589.81	$720	$870
6	$630	$631.13	$720	$840
7	$648.802	$671.809	$780	$930
8	$686.979	$704.017	$795	$930
9	$777.588	$790.688	$825	$870
10	$902.793	$920.964	$1005	$1110

Table 9 Practitioner's Cost and Health Care Agency's Payment at Different Group

4.5.3 Implementation Consideration

Since the practitioners depart and return to different places and are moving around the city when performing the schedule, online auctions, which break down the physical limitations, allow practitioners to enter and place bids at any time and any place before the auction ends. To spare practitioners the trouble of continuously monitoring the bidding process and repeatedly placing their bids, online auctions also allow practitioners to directly provide their necessary personal information to an automated bidding agency (called a proxy agency) which bids on their behalf. In the proposed iterative auction for HVS, the proxy agency will be used to generate and manage a set of feasible practitioner schedules and decide which schedule to submit, at which round, and at what price. Thus, the practitioner should provide the basic information, such as availability, location, preferences, and other needed information. In the meantime, the agency is supposed to collect and organize visit requests. Besides this, the agency should also be equipped with the algorithm to update bidding prices and select the payoff maximization schedule during the bidding process. If a practitioner prefers, the agency can also inform them regarding their bidding status and allow the practitioner to update their bids before the auction ends. For easy access, practitioners may install the proxy agency on a personal computer, a smart phone, or other mobile devices. Many online auctions provide a “buy it now” option to accommodate those buyers who cannot wait until the auction ends. A buyer can purchase the item immediately by paying the buy-it-now price. However, the buy-it-now price is usually a regular retail price, which can be much higher than the final auction price. Rigorously, we will not consider buy-it-now as part of the auction design.

4.6 Summary

This chapter describes a multi-agent scheduling system used to solve the HVS problem in a decentralized environment. The solution facilitates multilateral negotiation between practitioners to reduce health care agency costs. The proposed iterative bidding mechanism promises reduced computation by the HC Agency, while at the same time the experimental result shows the framework achieves a high efficiency of optimal solutions.

The designed system focuses on incoming visit request allocation, while it can handle the cancellation by adding a complementary functionality. The complete solution is expanded based on this system, in which the dynamic characteristic is partially reflected in the design. In the next chapter, we develop a simulation model that evaluates the scheduling system in a dynamic environment. The model simulates the operation of practitioners and the scheduling process of accommodating unexpected situations during practitioners’ performance of their schedules.

Chapter 5 Dynamic Scheduling for HVS

Having previously solved scheduling problems in a decentralized environment, this chapter proposes a dynamic approach to further accommodate dynamic changes. In practice, there are always unexpected situations occurring in home care delivery, such as new visit requests or cancellations, causing the initial schedule to quickly become obsolete. At some point in the active scheduled period, rescheduling will inevitably be needed to accommodate these unexpected changes. In this chapter, we enhance the designed decentralized scheduling algorithm to be able to deal with dynamic events. A simulation model is designed to support the decision making of the health care agency. The dynamic scheduling algorithm is implemented in this model to evaluate its effectiveness in a dynamic environment.

5.1 Dynamic Scheduling through Periodic Repair

In this section, we introduce a dynamic algorithm and select an appropriate modeling methodology. The proposed static scheduling algorithm is treated as a meta-algorithm in the dynamic setting that is called on repeatedly over the course of the operative schedule. The dynamic scheduling algorithm takes all scheduling-related information known at the moment of calling as its input, which is a different approach from that of the static algorithm.

We propose a periodic repair algorithm as the dynamic scheduling approach. This approach consists of two stages, which together generate an initial schedule in the first stage and then repair the schedule as needed in the second stage. In the first stage, the initial practitioner schedules are created by the agent-based scheduling system. The launch time of the first stage will be postponed as close as possible to the schedule's commencement for two reasons. Taking a weekly plan, for example, more scheduling-related information will be revealed as the start

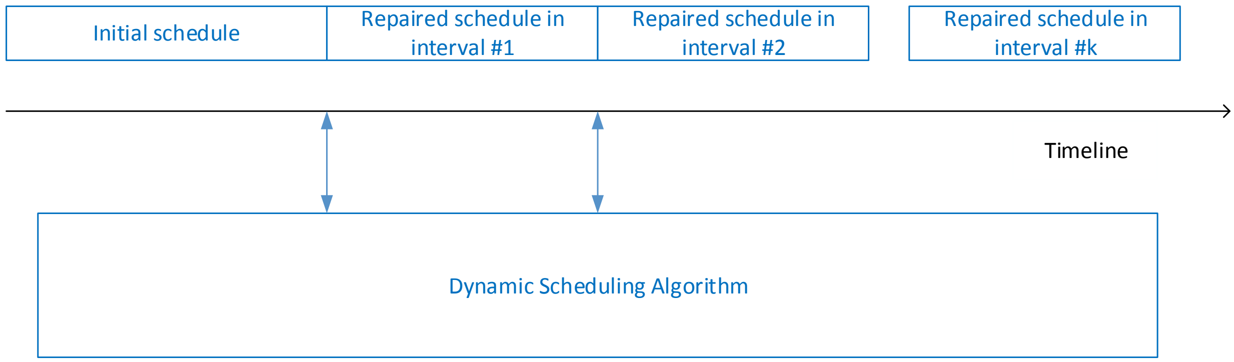

Figure 5 Dynamic Scheduling Flowchart

date of the new schedule draws closer, enabling the optimization to produce more efficient results due to the increased number of bidding objectives. Secondly, the contractual relationship between the practitioner and the required home visit also cannot be broken unilaterally from the practitioner's side in the second stage. Changes in the initial plan due to rescheduling will therefore be minimized through shortening the rescheduling time horizon.

In the second stage, the initial schedule will be repaired periodically as needed. Rescheduling repeatedly uses the decentralized algorithm as a meta-algorithm. Compared to static algorithms, which take the given information as their input, dynamic scheduling takes current scheduling related information as input and uses it to compute feasible packages. Each time it does so the system collects the incoming visit requests, the availability of practitioners, and the current practitioner schedules at the moment of rescheduling. As shown in Fig. 5, a practitioner's initial schedule is first created, followed by periodically repaired schedules generated from calling on the dynamic scheduling algorithm within the time horizon, allowing for a flexible period of repair. Theoretically, a large rescheduling interval brings better solutions, since, as mentioned above, the greater information regarding new visit requests allows for a more optimal solution compared to solutions with less information. However, the settled schedule also has to be delivered within a certain time before it starts to leave sufficient routing time. The interval setting has to take such scenarios into consideration. Due to the fact that visit requests arrive stochastically in practice, it's hard to decide in an analytical way what the interval should be. In

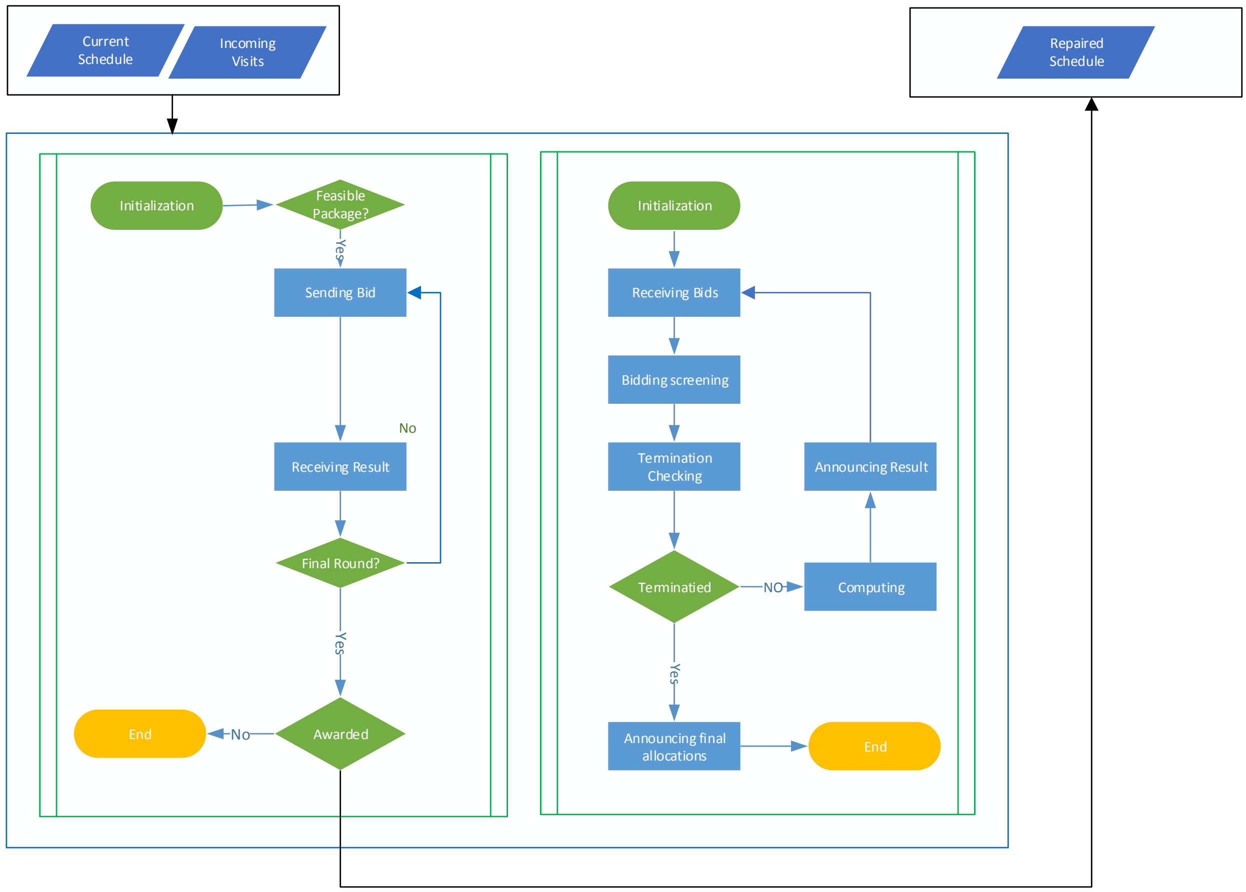

Figure 6 Dynamic Scheduling Workflow

light of this, a simulation model is developed in this chapter to aid the health care agency in making such decisions.

Fig.6 illustrates the workflows of both health care agency and practitioners when calling on the dynamic scheduling algorithm. When the algorithm is called, the registered practitioners compute their feasible packages based on their current availability derived from a real time schedule. The practitioners participate in the bidding if there are feasible packages for them. All registered practitioners report such information to the health care agency. Then the algorithm goes through the proposed bidding process, in which the health care agency plays the role of mediator to coordinate negotiations between practitioners. Finally, the awarded practitioners update their own schedules (Table 10).

```
Algorithm 3 dynamicScheduling
Input: registered practitioners RPS, current schedules CSS, incoming visits IVS
Output: revised schedules RSS
  while IVS ≠ ∅ do
    for all RPS do
    if feasible packages ≠ ∅ then
      PRPS ← RPS who participates bidding
    end if
    bidding start
    while bidding is not terminated do
      for all PRPS do
        create bid
        send bid
      end for
      health care agency checks termination condition
      health care agency computes and announces result
    end while
    health care agency announces final allocations
    for all PRPS do
      if the practitioner is awarded then
        RSS ← update its schedule
      end if
    end for
    return RSS
  end while
```

Table 10 Dynamic Scheduling Algorithm

5.2 Evaluating the Dynamic Scheduling Algorithm through Simulation

Abstracting a model from real word problems is a common way to evaluate alternative solutions, especially for huge and complex problems. The advance of computational power and software tools has made modeling and computer simulation much easier, with lower risks and costs. Analytical modeling, typically based on formulas and static dependencies, is used in various areas. In terms of the HVS problem, we have designed an agent-based system based on an analytical model in a static environment. However, a large number of problems either do not have any analytic solution or have very difficult solutions due to the influence of dynamic behaviors and events, such as the unexpected incoming visits, cancellations, or practitioner availability changes, all of which are typical of the HVS problem. In such scenarios, simulation modeling is a useful method to try when other modeling methods fail. Simulation predicts the performance of the model through observing the outcomes from uncertain inputs.

To date, there are three simulation methods to select from based on the abstraction level, namely system dynamics, discrete event modeling, and agent-based modeling. The critical decision to be made is choosing the proper abstraction level before modeling. Periodic reconsideration of the abstraction level during the development process is also typically needed.

Depending on the simulation project's goals and the nature of the problem, different problems may call for different methods or their combinations, a process called the multi-method model.

System dynamics is a method used to study dynamic systems in which time and geometrical dependencies are important. Discrete event modeling is a method based on a process, i.e., a sequence of operations being performed across entities. Agent-based modeling is a more recent method, which is normally triggered by a scenario in which a concrete process flow is not easily captured as a whole, even though the behaviors of individual entities are in fact known. It can be used at any level of abstraction.

Agent-based scheduling systems are based on a predesigned mechanism consisting of predefined operations and interactions between entities. Unexpected situations derived from the HVS problem are easily modeled as discrete time events in our model. Moreover, the agent-based scheduling system intuitively calls for agent-based modeling. Practitioners and health care agencies can be modeled as agents. Practitioner agents are considered as self-interested agents who make their own decisions independently. A decision-making process can be programmed into each practitioner entity. The Health Care Agency can be treated as another agent in this system, playing the role of a dispatcher who collects visit requests and performs dispatching. These two agents are put into a dynamic environment where discrete events are generated along with their particular time intervals, such that dynamic scheduling can be simulated. In the next section, we describe the design of the simulation model and how we implement the scheduling algorithm.

5.3 Simulation Model Design

We used Anylogic as the tool to implement our simulation model, as it is the only tool that brings together three different simulation methods within one modeling language (http://www.anylogic.com/). Anylogic is used extensively in the supply chain, logistics, manufacturing, business processes, health care, and marketing fields. A related example is that of Merkuryeva and Bolshakovs, who developed a simulation model of the Vehicle Routing Problem with Time Windows (VRPTW) in an Anylogic simulation environment [61]. The developed model, serves as an aid to analysts' decision-making, simulating goods deliveries from distribution centers to a set of shops.

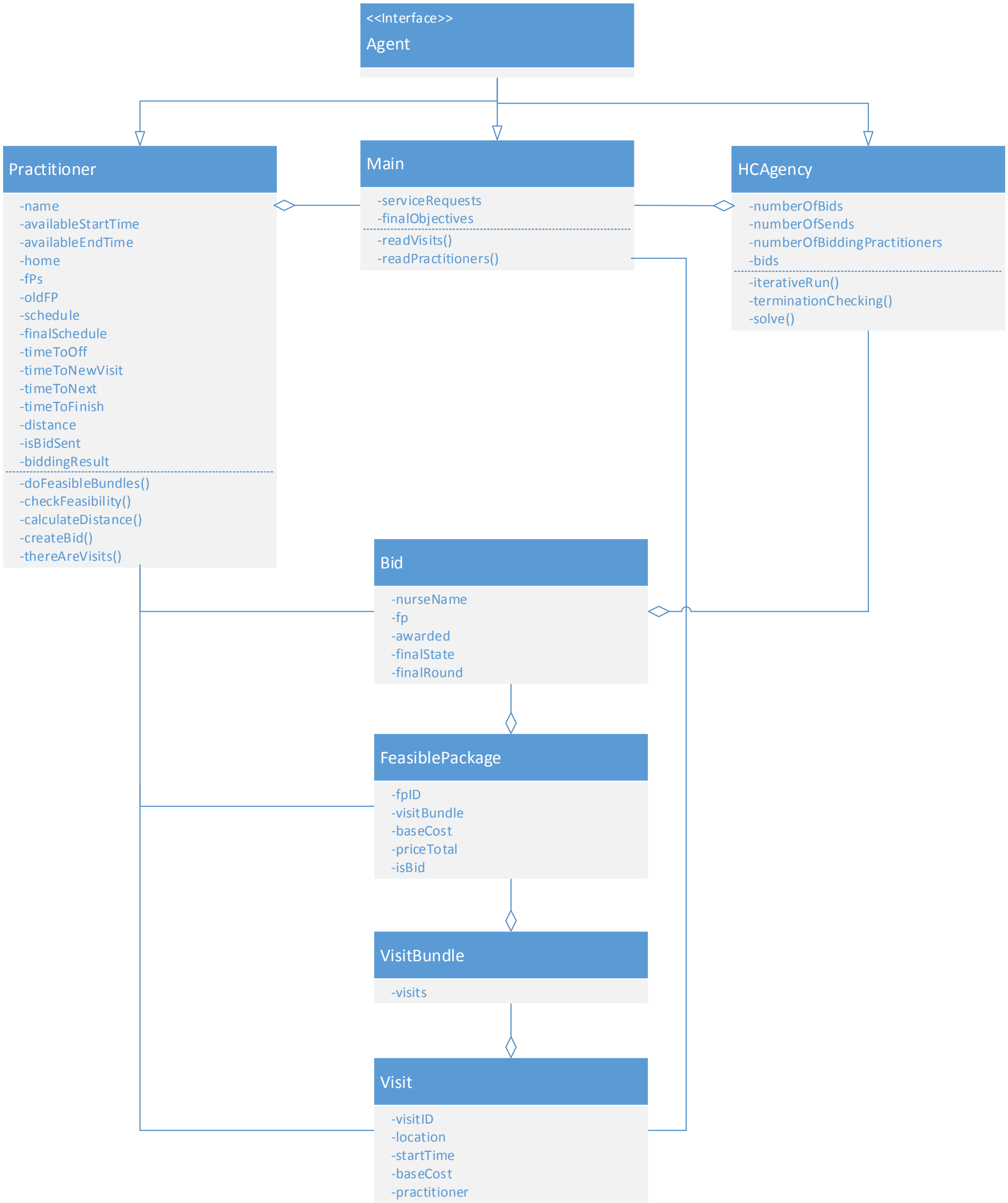

Figure 7 Class Diagram of Simulation Model

5.3.1 Model Structure

We used both a class diagram and a sequence diagram to describe the model structure and the message passing between the agents.

- **Class Diagram**

The class diagram illustrates the structure of the conceptual simulation model based on an Anylogic development environment by listing the essential classes, their attributes, methods, and the relationships among the objects.

Anylogic provides object interfaces to facilitate the modeling process. The Agent interface is the main component, which allows for the implementation of not only an agent but also their state or behavior. The agents can communicate with each other through calling functions. The simulation model contains two agents (active objects) and several classes. The *Main* object is the simulation model itself, which constructs the foundation and operational environment, in which both the HC Agency and Practitioner Agents are aggregated. The classes Bid, FeasiblePackage, VisitBundle, and Visit encapsulate their own properties and are used by all agents. Details about each active object are introduced in the following sections.

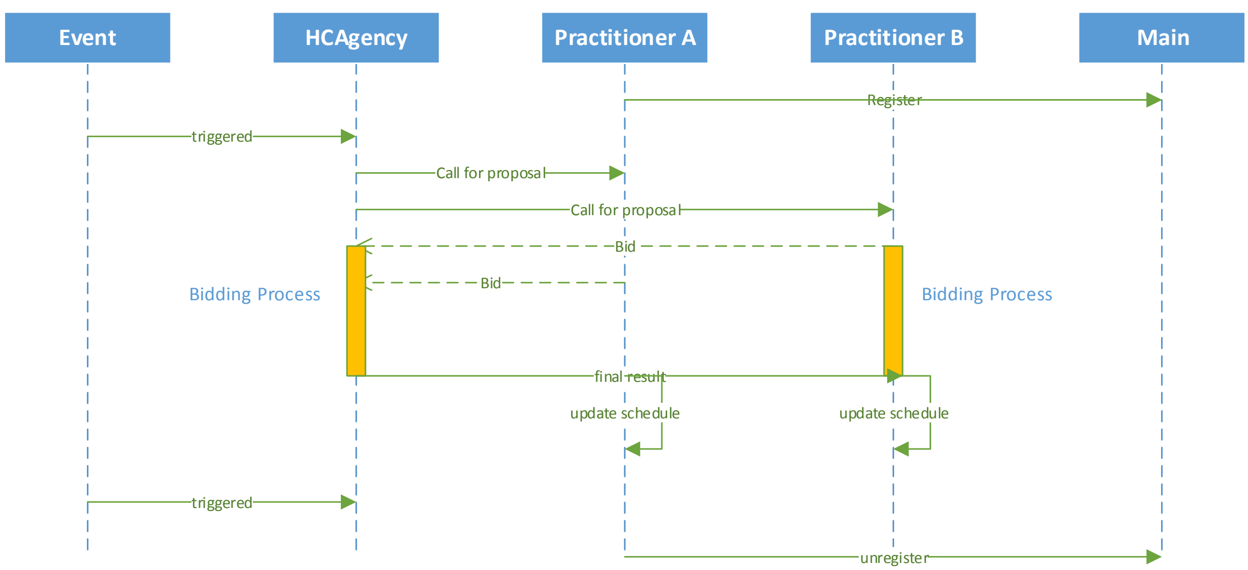

Figure 8 Sequence Diagram of Simulation Model

- **Sequence Diagram**

The sequence diagram shows essential communications between agents. The Main object serves as a facilitator and accepts the register/unregister of Practitioner Agents. The HC Agency sends call for proposal messages to the Practitioner Agent group whenever necessary. The Practitioner Agents then respond to the message by sending back bids. The bidding process iterates until the termination condition is satisfied. After the process, the bidding result is sent back to each Practitioner Agent who updates their schedules accordingly.

5.3.2 Running Environment

Our study focuses on resource allocation, which reflects the rostering characteristics of the HVS problem, while the simulation model also involves the routing aspect. In the *Main* object, we collect map information from a GIS map downloaded from an online map service provider (https://www.mapquest.com/). We employ the Anylogic routing server to generate routes for practitioners, the main purpose of which is to ascertain the distance and travel time for each visit at a predefined travel speed. Using this component of Anylogic, the simulation model comes very close to the real world problem by taking the real map and routing data as inputs. There are

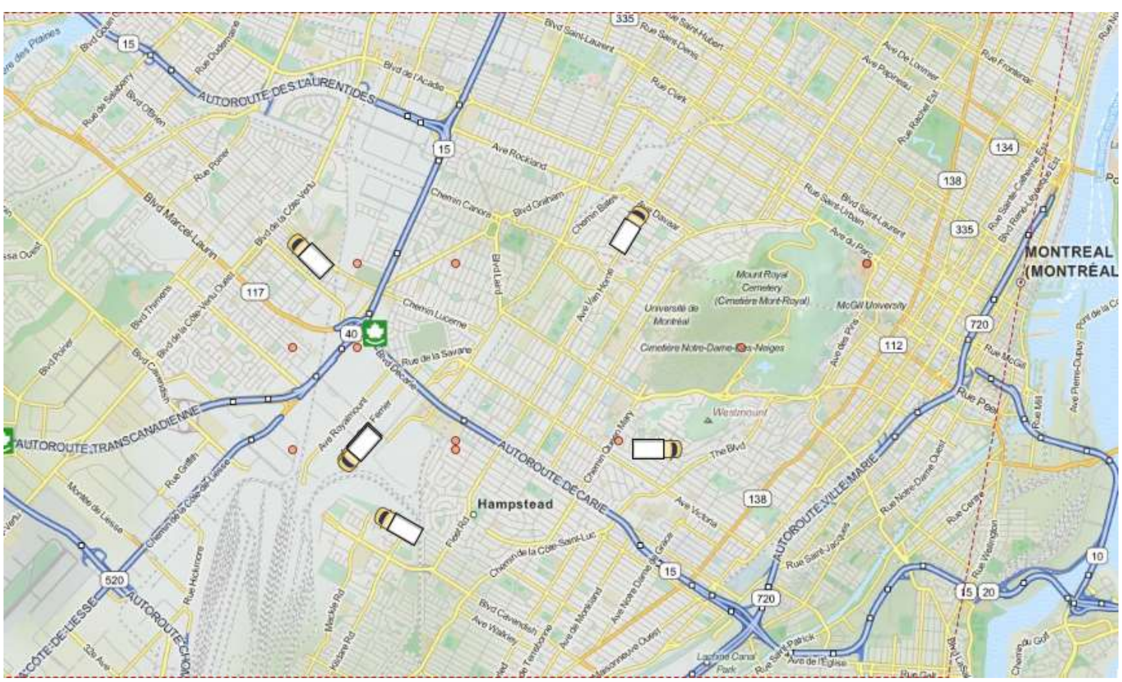

Figure 9 Running Environment of Simulation Model

a few alternative routing methods, such as the fastest or shortest. And multiple road types are available, including car, rail, bike and foot. We selected the fastest routing method and chose car as the transportation tool, while the details about the beneath routing algorithm are not our concern in this thesis.

The above figure shows the practitioner's locations and existing home visits on a map of the Montreal area. The truck symbol represents the Practitioner Agent and the red points indicate home visit requests. Practitioners travel through the visits according to their own schedule, following the routes navigated by the routing server. The schedules are dynamically updated according to the embedded scheduling algorithm as the simulation engine generates discrete time events. The new incoming visit locations will be displayed on the map as well.

The discrete time event generator is setup in *Main* object. The event trigger type could be timeout, rate, or condition. Rate triggered events are frequently used to model independent event arrivals. Such events are triggered periodically, with time intervals distributed exponentially within the parameter, i.e., if the rate is set to 1, the event will occur on average 1 time per time unit. Timeout trigger type is used to generate events after a certain time unit. And condition type

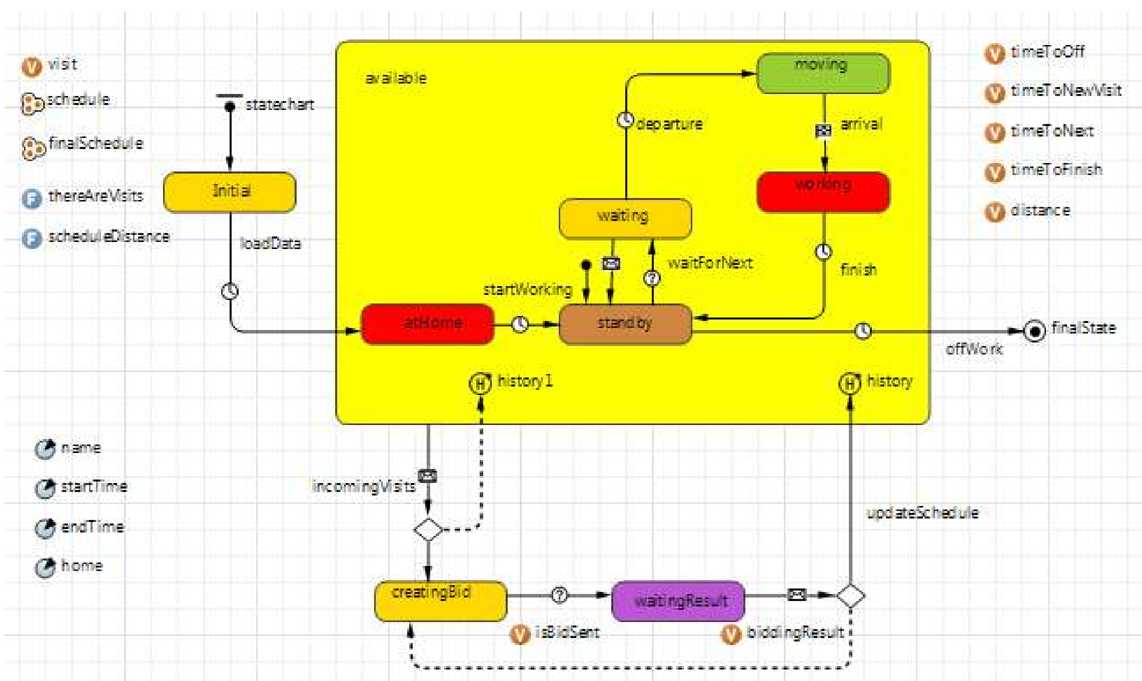

Figure 10 Practitioner Agent State Chart

allows users to configure the trigger conditions of the events. Rate event trigger type is used in this model, and a random number of visits are created each time the event is triggered.

The profile of practitioners and the corresponding planned schedules are initialized with the simulation startup. The planned schedule could be empty or could contain some existing visits. The Health Care Agent's decision-making requires an optimization solver to find the optimal allocation, in which the integer-programming model is programmed (http://www.opttek.com/OptQuest).

5.3.3 State Chart of Practitioner Agent

The Practitioner Agent is another active object in this model. These agents go through the visits, participate in the bidding and maintain their own schedules. The parameters are initialized as the preference information of practitioners at the beginning of the simulation. Some essential variables are declared in order to store dynamical values during the simulation, including a list of dynamic schedules, a list of final schedules, a list of feasible packages, and a variable for the bids from the previous round of the bidding process. The dynamic schedule contains only the upcoming visits at the moment. Completed and cancelled visits are removed from this list and new allocated visits are added dynamically. Another schedule contains all the actually performed visits. The feasible package will be stored in advance each time a call for proposal is received, and will be cleaned after the bidding is terminated. The last variable, the previous bidding item, is used to calculate utility in next round in the auction.

Object Practitioner simulates both the schedule performing process and the bidding process, which are described by the state chart in Fig. 12. It also defines the states and transitions between these two processes. The Practitioner Agent's states include a composite state named *available* which consists of *atHome, standby, waiting, moving* and *working*. An *initial* state and a *final* state are used for variable initialization and statistical calculation. The states and trigger conditions are described in detail as follows:

1) The *initial* state is for the Practitioner Agent registering on a specific day. Registered practitioners are available to take visits, while they could either have pre-allocated visits or an empty schedule. The profile of a practitioner, including the name, availability and

home location, is initialized in this step. Visits are loaded into their schedule if there are any. In addition, the availability of practitioners has to be loaded in advance to guarantee the correctness of calculations depending on it. The trigger condition of *initial* state is a minimum double value in this state chart to allow for initiation.

2) An entry action is setup on the *atHome* state to initiate the location and move the Practitioner Agents to their departure and return locations. The states are fired to *Standby* when it's the time to start working.
3) The *Standby* state is supposed to interact with multiple states, including *atHome, waiting, working* and the *final* state. Entering the *standby* state from *atHome* means the agent is available from now on until their off-work time. A transition to the *waiting* state will be triggered if there are any remaining visits in their schedule, but the Practitioner Agent goes to the final state if they reach off-work time. The agent is unregistered at the time the latter transition is triggered and moves back to their home location. Besides the *atHome* state, there are another three incoming transitions: an *Initial State Pointer*, the *waiting* state and the *working* state. An *Initial State Pointer* is setup as the default entry of the *available* composite state to which the control is passed. The state will return from the *waiting* state to the *standby* state whenever there is a visit cancellation before the agent starts travelling towards that visit, while the visit is not allowed to be cancelled when the agent is on the way. Another incoming transition is from *working* state, which indicates the visit is finished.
4) Each time the control is passed to the *waiting* state, the system obtainss the current location and the earliest visit in its schedule, and calculates the travel time. Besides the transition directs to *standby*, another one directed to the *moving* state is triggered at the moment the agent has to depart to reach the next visit on time. Such a visit is removed from its dynamic schedule and added to the final schedule when such a transition is triggered.
5) The Practitioner Agent is moving to the upcoming scheduled visit when the system enters the *Moving* state. This state is triggered to the *working* state when they reach their destination.
6) After the pre-defined serving time of the specific visit, the system triggers the state back from *working* to *standby*.

7) The *final* state indicates the end of the available time of the Practitioner Agent, otherwise the agent stays in the *standby* state even if there are no upcoming visits in their dynamic schedule. The statistical calculation is conducted in this transition.

As the Practitioner Agent is performing their schedule, the call for proposal message will trigger the state out to a bidding process from whatever state the agent is in. After the HC Agency determines the allocation, the state of participating Practitioner Agents will return to the moment it was called out. The bidding process includes two states, *creatingBid* and *waitingResult*.

1) After receiving the call for proposal message from the HC Agency, the Practitioner Agent calculates the utility and generates feasible bundles (*doFeasibleBundles()*). If the agent does not have any feasible bundles, it returns to its state immediately. Otherwise, the agent participates in the bidding and enters the *creatingBid* state (*createBid()*). At this time, the agent will send an "UPDATE" message to the HC Agent. After the agent sends their bid to the HC Agency, they enter the *waitingResult* state.
2) After receiving the bidding result, the agent updates their schedule accordingly and returns to the state where it was triggered if this is the final bidding round. Otherwise, it continues to the next round and the state is transited back to *creatingBid*.

5.3.4 State Chart of Health Care Agency

The object HC Agent plays the role of dispatcher in the model. The HC agent is an autonomous agent who calls for proposal messages, solves the optimization problem, and sends the bidding result back to Practitioner Agents. A bidirectional connection is setup between Practitioner Agents and the HC Agency for message passing. The solicited bids sent from Practitioner Agents are stored into a variable bids, which is one of the two inputs of the optimization model. Two other variables, *numberOfBids* and *numberofSends*, are setup to count the number of received bids and the times at which the result is sent back to the Practitioner Agents, respectively. The HC Agency monitors the number of received bids to decide when the winner determination problem should be solved. Similarly, the HC Agency goes back to the *waitingBids* state after all results are sent.

The HC Agency utilizes a state chart to control its behaviors (Fig. 13). Three states are defined in this chart: *initial, waitingBids* and *calculating.*

1) The *initial* state will be triggered at the same time as the *initial* state of the Practitioner Agent, which leaves time for parameter initiation. Thereafter, the HC Agency moves to the *waitingBids* state, at the same time as the Practitioner Agents start to register. The trigger condition of *initial* state is a minimum double value.

2) In the *waitingBids* state, a transition directed to itself will be triggered by the message "UPDATE" from a Practitioner Agent. Each time a bid is sent, the practitioner sends an update message to the HC Agency to update the state chart-related data. The reason we setup such a mechanism is that the values pertaining to the state chart (e.g., the *numberOfBids*) will not otherwise be updated automatically. The other outgoing transition will be triggered to the *calculating* state by the appropriate condition, which is when the number of received bids equals the number of registered practitioners, at the same time as the number of received bids is NON-zero. This transition solves the allocation problem, which means it starts the bidding process we have proposed. As we described previously, the backup practitioners will always participate in the bidding with the standard price and no price decreases. Thus, the bids of such practitioners are known, as soon as the new home visit information is decided. The HC Agency adds those bids to

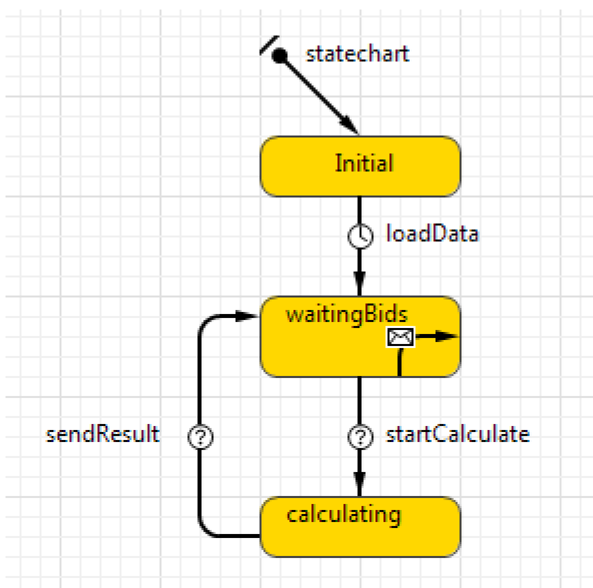

Figure 11 Health Care Agency State Chart

its collection in one step in this action. The HC Agency solves the winner determination problem by passing the set of bids and the set of incoming visits to the optimization model, and obtains the results from the solver. According to the result, the HC Agency checks the termination condition, then either sends a modified call for proposal message in the event that the bidding is not supposed to be terminated, or notifies the Practitioner Agents of the final allocation. After the bidding process is complete, the *bids* will be cleaned. The number of results sent to Practitioner Agents will be setup to zero when this transition is triggered.

3) After the bidding results are sent to all participated Practitioner Agents, the system triggers the transition that directs it from *calculating* to *waitingBids*. At the same time, the number of received bids will be set to zero.

5.4 Simulation Model Implementation

This section summarizes the algorithms of the essential functions, all of which are implemented in Java.

5.4.1 Feasibility Checking

```
Algorithm 3 checkFeasibility
Input: Visit visit
Output: Boolean
  Boolean feasible
  currentVisit
  if service time inteval ∉ feasible time then
    feasible ← false
  else
    if schedule = ∅ then
      if finalSchedule ≠ ∅ then
        currentVisit ← last visit in finalSchedule
       else
        currentVisit ← home
       end if
       if time(currentVisit,visit)<= start time of visit then
        feasible ← true
       else
        feasible ← false
       end if
     else
       if start time of visit < start time of first visit in schedule then
```

```
        if finalSchedule ≠ ∅ then
          currentVisit ← last visit in finalSchedule
        else
          currentVisit ← home
        end if
        if time(currentVisit,v)<= start time of visit and time(v,first visit in
schedule)<= start time of first visit in schedule then
          feasible ← true
        else
          feasible ← false
        end if
      else if start time of visit >= end time of last visit in schedule then
        if time(last visit in schedule,visit)<= start time of visit then
            feasible = true
          else
            feasible = false
          end if
      else
        if time(previous visit in schedule, visit)<=start time of visit and
           time(visit, next visit in schedule)<=start time of next visit then
          feasible ← true
        else
          feasible ← false
        end if
      end if
    end if
  end if
return feasible
```

Table 11 Feasibility Checking Algorithm

There are several different important issues to consider when checking the feasibility of a visit: 1) whether the service time is in the feasible time range of a practitioner or not; 2) whether there are upcoming visits in a practitioner's dynamic schedule; 3) whether the incoming visit would be first in the practitioner's dynamic schedule should they accept it; 4) whether the incoming visit would be last in their dynamic schedule; and 5) whether the visit time of the incoming visit is feasible between already scheduled visits. We apply the same checking rule for the above-mentioned scenarios, which is that all the time constraints should be satisfied, including the visit start time, as well as the availability of the practitioner. The travel time is calculated based on the route given by the routing server and the predefined travel speed.

5.4.2 Distance Calculation

```
Algorithm 4 calculateDistance
Input: List<Visit> schedule
Output: totalDistance
  totalDistance ← 0
  if schedule ≠ ∅ then
    if finalSchedule ≠ ∅ then
      previous ← last visit in finalSchedule
    else
```

```
      previous ← home
    end if
    departDistance ← distance(previous,the first visit in schedule)
    returnDistance ← distance(the last visit in schedule, home)
    for Each Visit in schedule do
      scheduleDistance ← scheduleDistance + distance(v_i,v_{i+1})
    end for
    totoalDistance ← scheduleDistance + departDistance + returnDistance
  end if
  return totalDistance
```

Table 12 Distance Calculation Algorithm

This algorithm is used for calculating the overall travel distance of a schedule. The input is the dynamic schedule at the moment such a method is invoked. The overall distance is defined as the total distance for the practitioner departing from their current location, finishing all the visits on their schedule, and finally, returning home.

5.4.3 Feasible Package Generation

```
Algorithm 5 doFeasibleBundles
Input: schedule, objectives
Output: List<FeasiblePackage>
  Boolean breakIndicator
  currentDistance ← calculateDistance(schedule)
  for Each VisitBundle vb in objectives do
    breakIndicator ← false
    for Each Visit visit in vb do
      if checkFeasibility(visit) = true then
        continue
      else
        breakIndicator ← true
        break
      end if
    end for
    if breakIndicator = true then
      continue
    end if
    newSchedule ← add vb to schedule
    afterDistance ← calculateDistance(newSchedule)
    baseCost ← (afterDistance – currentDistance) * millage rate
    priceTotal ← size of vb * standard price
    if priceTotal – baseCost > 0 do
      create a FeasiblePackage fp
      add fp to fPs
    end if
end for
return fPs
```

Table 13 Feasible Package Generation Algorithm

doFeasibleBundles creates the feasible bundles for each practitioner. The feasible bundle is that which has passed feasibility checking and has a positive utility for the practitioner. The travel cost is scaled to a money value through multiplying a mileage rate by the travel distance. The overall price of a bundle is the sum of a standard price per visit. *VisitBundle* is an object that contains one or more *Visit* objects. *FeasiblePackage* is an object that contains a *VisitBundle* object and the bundle's *baseCost* and *priceTotal*.

5.4.4 Bid Creation

```
Algorithm 6 createBid
Input: epsilon
Output: Bid
  if oldFP ≠ ∅ then
    for Each FeasiblePackage fp in fPs do
      if oldFP = fp then
        price of fp ← price of fp - epsilon
        oldFP ← fp
      end if
    end for
  end if
  maxUtility ← 0
  FeasiblePackage maxUtilityFP
  for Each FeasiblePackage fp in fPs do
    utility← price - cost
    if utilty > maxUtility then
      maxUtilityFP ← fp
      maxUtility ← utility
    end if
  end for
  if maxUtility ≠ 0 then
    numberOfBids ++
    oldFP ← maxUtilityFP
    isInFinalState ← false
  else
    isInFinalState ← true
    for Each FeasiblePackage fp in fPs do
      if oldFP = fp then
        price of fp ← price of fp + epsilon
        maxUtilityFP ← fp
      end if
    end for
    oldFP ← maxUtilityFP
  end if
  create new bid with maxUtilityFP and isInFinalState
  Attach other needed information to bid
  return bid
```

Table 14 Bid Creation Algorithm

Only in the case where there are feasible packages for the practitioner, who is also able to participate in the bidding, is the *createBid* method invoked. *oldFP* is a *FeasiblePackage* object which is initialized outside the method as null. In the first round of bidding, the bidding price is the standard price of such a *VisitBundle*. The bidding price decreases by the epsilon starting from the second round. Thereafter, the feasible package with maximum utility will be selected as the bidding objective. The final state is determined through checking the value of *maxUtility*. Finally, the bid is created based on such objectives and the bidding price. The practitioner's identification and other required information should be attached to the bid, such as the indicator of the final state, final round, and awarded.

5.4.5 Termination Checking

```
Algorithm 7 terminationChecking
Input: bids
Output: Boolean
  terminate ← true
  if numberOfBids = 1 then
    return terminate
  else
    for Each Bid bid in bids do
      if bid is not in final state then
        terminate ← false
        break
      end if
    end for
    return terminate
  end if
```

Table 15 Termination Checking Algorithm

At each bidding round, the auctioneer will check whether the bidding process needs to be terminated. In cases where there is only one bid, the bidding is terminated by simply awarding the visit to that bid. Otherwise, the bidding is terminated only when every bid is in its final state. The final state indicator in each bid is assigned as either true or false in *createBid()*.

5.4.6 Iterative Bidding

```
Algorithm 8 iterativeRun
Input: serviceRequests, bids
Output: Void
  for Each Visit visit in serviceRequests do
    create bid for visit with standard price
    add bid to bids
  end for
  call solver
  if terminationChecking() = false then
    call for proposal
  else
```

```
    for Each Bid bid in bids do
      set bid to final round
      send bidding result
      numberOfSends ++
    end for
  end if
```

Table 16 Iterative Bidding Algorithm

Based on the assumption that there are always backup practitioners willing to take a single visit at standard price, the model creates bids for those practitioners and adds them to the collection of bids in each round of bidding. The optimization solver in which the model is programmed is called upon when the model input is ready, and the awarded indicator in the bid is set to true depending on the computing result. Thereafter, additional calls for proposal are sent whenever termination-checking fails; otherwise the final round indicator is set to true and bidding is terminated.

5.4.7 Optimization Model for Winner Determination Problem

The optimization model is programmed with the OptQuest optimization solver embedded in Anylogic. The mathematical model can be found in section 4.2.1.4, while the code itself is shown in Fig. 17.

```
try {
        // Create Engine
        Engine engine = createEngine();
        // Set stop time, initialize random number generator:
        engine.setStopTime(50);
        engine.setDefaultRandomGenerator(new Random());
        // Create optimization engine
        final COptQuestOptimization opt = ExperimentOptimization.createOptimization(engine);
        // Set optimization variable
        ArrayList<COptQuestVariable> v = new ArrayList<COptQuestVariable>();
        for (int i=0; i<dispatcher.bids.size(); i++) {
              final COptQuestVariable x = new COptQuestBinaryVariable();
              v.add(x);
        }
        for (int i=0; i<dispatcher.bids.size(); i++) {
              opt.AddVariable(v.get(i));
        }

        // Create objective
        final COptQuestObjectiveFunction obj = new COptQuestObjectiveFunction();
        obj.SetMinimize();
              for (int i=0; i<dispatcher.bids.size(); i++) {
               obj.AddVariable(v.get(i),dispatcher.bids.get(i).getFp().getPriceTotal());
        }
        opt.AddObjective(obj);
        // Create constraint
```

```
        for (Visit visit : main.serviceRequests) {
          final COptQuestEQConstraint constraint = new COptQuestEQConstraint();
            for (int i=0; i<dispatcher.bids.size(); i++) {
              if
(dispatcher.bids.get(i).getFp().getVisitBundle().getVisits().contains(visit) == true) {
                constraint.AddVariable(v.get(i),1);
              }
            }
               constraint.SetRHS(1);
               opt.AddConstraint(constraint);
        }
        // Set the number of iterations to run
        opt.SetMaximumIterations(50);

        // Perform optimization
        opt.Optimize();

        // Output results
        COptQuestSolution bestSolution = opt.GetBestSolution();

       for (int i=0; i<dispatcher.bids.size(); i++) {
              dispatcher.bids.get(i).setAwarded(bestSolution.GetVariableValue(v.get(i)));
       }

 } catch (COptQuestException e) {
        traceln(e.Description());

 }
```

Figure 12 Optimization Model for Winner Determination in Simulation

5.5 Simulation Scenarios and Results

The model simulates the dynamic scheduling process as well as practitioners' daily routines. The preferences of practitioners are loaded into the model before the simulation starts. Meanwhile, a set of planned visits for the specific day and their service start times, locations and the name of each assigned practitioner are imported into each practitioner's initial schedule. A group of practitioners can become available, along with the time going, according to the particular time they each start working. They depart from their individual homes, visit their patients, and end their day once they return. Home visit requests will occur randomly over the course of their working day. The health care agency assigns practitioner to incoming visits following the rules of the designed scheduling mechanism. The model time unit is set up to hours and the simulation starts from time zero. This section contains four scenarios that describe possible cases during dynamic scheduling.

- **Scenario 1**

In this scenario, a simple example can be used to illustrate the dynamic scheduling process. Since the performance of the scheduling mechanism has already been evaluated in the last chapter, we keep the example here simple enough to present the dynamic characteristic of the problem. Table 17 lists the scheduled visit information and the value of their attributes. For example, visit #101 is located at the listed coordinates and practitioner Lily is supposed to visit there at time 2 and finish service at time 2.5. Lily will reach this location exactly at time 2 from either her home or from a previous visit, depending on her schedule. Practitioner profiles are listed in Table 18, including their names, availabilities, and the location coordinates each practitioner. For example, Mayo is available from time 2 to time 10, and is expected to depart from and return to the location of the corresponding coordinates. The simulation model loads this information and constructs the initial schedules of each practitioner. The initial schedules are listed in Table 19. Each time a dynamic event is triggered, one or more new visits with initialized attributes are created as the bidding objectives. In this example, the standard price of a single visit is set to $30, the epsilon is set to 10 and the mileage cost is set to $2. We suppose the patients have to make the appointment at least 2 hours in advance. The trigger rate of the discrete events is set to an average once every two hours.

Three events are triggered and a total of four visits are generated in this simulation run. The allocation results and taking prices are listed in Table 20. The last three columns in the table are the payment from the proposed solution, the payment of First-Come-First-Serve, and the practitioner's cost of taking the visit. Compared to the manual scheduling method, which follows FCFS, we can observe that the overall payment is reduced by 50%. Incoming visits are assigned and the corresponding schedules are updated, as shown in the final schedule (Table 21).

visitID	homeX	homeY	startTime	endTime	Practitioner
101	45.5011	-73.6015	2	2.5	Lily
102	45.4888	-73.6777	4	4.5	Lily
103	45.5111	-73.58	6.5	7	Jenny
104	45.5011	-73.6777	3	3.5	Mayo
105	45.5111	-73.58	5	5.5	Mayo
106	45.5111	-73.65	3	3.5	Ciara

Table 17 Planned Visits in Run 1

Name	startTime	endTime	homeX	homeY
Jenny	6.00	10.00	45.51223	-73.67496
Lily	1.00	8.00	45.52	-73.62
Mayo	2.00	10.00	45.4888	-73.61539
Ciara	2.00	10.00	45.4887	-73.67
Emily	1.00	10.00	45.48	-73.66

Table 18 Profile of Practitioners

Practitioner	Schedule
Lily	{Visit#101, startTime:2.0} {Visit#102, startTime:4.0}
Mayo	{Visit#104, startTime:3.0} {Visit#105, startTime:5.0}
Emily	
Jenny	{Visit#103, startTime:6.5}
Ciara	{Visit#106, startTime:3.0}

Table 19 Initial Schedules

Event Time	visitID	startTime	endTime	Practitioner	Payment	FCFS	Cost
3.143	1	5.93	6.43	Lily	10	30	2.01
4.088	2	8.92	9.42	Jenny	10	30	8.75
6.898	3	8.96	4.88	Emily	20	30	19.31
6.898	4	9.0	5.27	Mayo	20	30	19.7

Table 20 Dynamic Scheduling Result in Run 1

Practitioner	Schedule
Lily	{Visit#101, startTime:2.0} {Visit#102, startTime:4.0} **{Visit#1, startTime:5.933}**
Mayo	{Visit#104, startTime:3.0} {Visit#105, startTime:5.0} **{Visit#4, startTime:8.998}**
Emily	**{Visit#3, startTime:8.965}**
Jenny	{Visit#103, startTime:6.5} **{Visit#2, startTime:8.924}**
Ciara	{Visit#106, startTime:3.0}

Table 21 Final Schedules in Run 1

The event occur rate, practitioner number, and epsilon of bidding can all be adjusted as needed in order to evaluate the model's performance in different scenarios. We set up another three scenarios that may reflect the business development process.

- **Scenario 2**

The second scenario assumes that the home visit requests are increased in both the initial schedules and later on. In this run, three more planned visits are added into the initial schedules, and a total of nine incoming visits occur during the performing time (the discrete time event rate is 0.6). As illustrated in Table 24, while most visits are assigned to practitioners, there are some special cases:

1) Visit#9 cannot be taken because its finish time falls outside the working time for all nurses. Visit#11 needs to be handled manually because no working practitioner is able to take this visit due to its lack of feasibility.
2) Visit #8 and visit#10 did not reach the lowest possible price, as there is only one practitioner who is able to take those visits, meaning there were no competitors in their bidding. These visits are assigned to the practitioner directly.

visitID	homeX	homeY	startTime	endTime	Practitioner
107	45.503	-73.66	7	7.5	Ciara
108	45.499	-73.599	5	5.5	Emily
109	45.505	-73.62	8	8.5	Emily

Table 22 New Planned Visits in Run 2

Practitioner	**Schedule**
Lily	{Visit#101, startTime:2.0} {Visit#102, startTime:4.0}
Mayo	{Visit#104, startTime:3.0} {Visit#105, startTime:5.0} **{Visit#6, startTime:8.956}**
Emily	{Visit#108, startTime:5.0} **{Visit#1, startTime:7.005}** {Visit#109, startTime:8.0} **{Visit#7, startTime:8.938}**
Jenny	{Visit#103, startTime:6.5} **{Visit#2, startTime:7.35}** **{Visit#5, startTime:8.142} {Visit#8, startTime:9.191}**
Ciara	{Visit#106, startTime:3.0} {Visit#107, startTime:7.0} **{Visit#3, startTime:7.799}** **{Visit#4, startTime:8.523} {Visit#10, startTime:9.322}**

Table 23 Final Schedules in Run 2

Event Time	**visitID**	**startTime**	**endTime**	**Practitioner**	**Price**	**FCFS**	**Travel Cost**
2.619	1	7	7.5	Jenny	10	30	7.65
2.619	2	7.35	7.85	Emily	20	30	14.97
5.434	3	7.8	8.3	Ciara	20	30	12.75
5.629	4	8.52	9.02	Ciara	30	30	21.15
5.629	5	8.14	8.64	Jenny	20	30	17.5
6.911	6	8.96	9.46	Mayo	20	30	18.86
6.911	7	8.94	9.44	Emily	10	30	4.58
7.465	8	9.19	9.69	Jenny	30	30	15.58
7.859	9	9.67	10.17	**N/A**	N/A	N/A	N/A
7.888	10	9.32	9.82	Ciara	30	30	8.76
7.888	11	9.7	10.2	**Manual**	30	30	30

Table 24 Dynamic Scheduling Result in Run 2

- **Scenario 3**

In this scenario, three more visits are added to the initial plan, and a total of 13 new visits are entered into the system (discrete event rate = 1). We analyze some special cases in this scenario as follows:

1. Visit #3 and visit #4 were awarded to Ciara as a package. This is one of the feasible packages for the practitioner, and the visits in this package will be added to Ciara's dynamic schedule once the bidding terminates.
2. More visits need to be handled manually by the health care agency compared with the last experiment run. In case the resources are static, the number of such cases would increase as the number of visit requests grows. The health care agency may make the decision to involve more practitioners according to the simulation results due to its own business requirements.

visitID	homeX	homeY	startTime	endTime	Practitioner
110	45.5111	-73.6	8	8.5	Jenny
111	45.503	-73.62	2	2.5	Emily
112	45.49	-73.58	7	7.5	Mayo

Table 25 New Planned Visits in Run 3

Practitioner	**Schedule**
Lily	{Visit#101, startTime:2.0} {Visit#102, startTime:4.0}
Mayo	{Visit#104, startTime:3.0} {Visit#105, startTime:5.0} {Visit#112, startTime:7.0} **{Visit#9, startTime:8.69}**
Emily	{Visit#111, startTime:2.0} {Visit#108, startTime:5.0} **{Visit#5, startTime:7.291}** {Visit#109, startTime:8.0} **{Visit#12, startTime:8.894}**
Jenny	{Visit#103, startTime:6.5} **{Visit#8, startTime:7.295}** {Visit#110, startTime:8.0} **{Visit#7, startTime:8.744}**
Ciara	{Visit#106, startTime:3.0} **{Visit#3, startTime:5.123} {Visit#1, startTime:6.187}** {Visit#107, startTime:7.0} **{Visit#6, startTime:7.595} {Visit#4, startTime:8.454} {Visit#13, startTime:9.263}**

Table 26 Final Schedules in Run 3

Event Time	**visitID**	**startTime**	**endTime**	**Practitioner**	**Price**	**FCFS**	**Travel Cost**
1.572	1	6.19	6.69	Ciara	20	30	18.1
1.572	2	4.68	5.18	**Manual**	30	30	30
2.044	3	5.12	5.62	Ciara	30	30	28.2
2.044	4	8.45	8.95			30	
2.044	5	7.29	7.79	Emily	30	30	21.33

4.166	6	7.59	8.09	Ciara	10	30	7.8
4.166	7	8.74	9.24	Jenny	10	30	9.83
4.844	8	7.3	7.8	Jenny	30	30	14.55
6.6	9	8.69	9.19	Mayo	10	30	6.38
6.617	10	8.86	9.36	**Manual**	30	30	30
6.617	11	8.7	9.2	**Manual**	30	30	30
6.617	12	8.89	9.39	Emily	30	30	14.46
7.292	13	9.26	9.76	Ciara	30	30	10.61

Table 27 Dynamic Scheduling Result in Run 3

- **Scenario 4**

Since the visit requests increased with the business development of the health care agency, it may want to add more resources in order to maintain its visit acceptance rate and service quality. In this scenario, two more practitioners are added, as listed in Table 29, while we keep all other parameters unchanged. The simulation result of this run shows the percentage of visits that need to be handled manually decreases as the available resources are increased.

Name	startTime	endTime	homeX	homeY
Anne	1.00	10.00	45.47	-73.66
Key	1.00	10.00	45.52	-73.62

Table 28 New added Practitioners in Run 4

Event Time	visitID	startTime	endTime	Practitioner	Price	FCFS	Travel Cost
1.048	1	4.11	4.61	Mayo	20	30	19.94
1.458	2	8.39	8.89	Ciara	30	30	13.14
3.632	3	7.33	7.83	Lily	10	30	3.73
3.633	4	7.08	7.58	Emily	20	30	18.3
3.633	5	8.87	9.37			30	
4.675	6	7.15	7.65	Key	20	30	10.03
4.675	7	7.2	7.7	Jenny	20	30	19.44
4.687	8	6.71	7.21	Lily	20	30	13.13
4.687	9	8.72	9.22	Jenny	10	30	7.56
4.918	10	8.63	9.13	Anne	30	30	27.42
4.918	11	7.92	8.42			30	
6.454	12	8.61	9.11	Mayo	30	30	19.28
6.79	13	8.95	9.45	**Manual**	30	30	30
6.79	14	8.85	9.35	Key	30	30	19.47

Table 29 Dynamic Allocation Result in Run 4

Practitioner	Schedule
Lily	{Visit#101, startTime:2.0} {Visit#102, startTime:4.0} **{Visit#8, startTime:6.707} {Visit#3, startTime:7.33}**
Mayo	{Visit#104, startTime:3.0} **{Visit#1, startTime:4.108}** {Visit#105, startTime:5.0} {Visit#112, startTime:7.0} **{Visit#12, startTime:8.607}**
Emily	{Visit#111, startTime:2.0} {Visit#108, startTime:5.0} **{Visit#4, startTime:7.077}** {Visit#109, startTime:8.0} **{Visit#5, startTime:8.873}**
Jenny	{Visit#103, startTime:6.5} **{Visit#7, startTime:7.198}** {Visit#110, startTime:8.0} **{Visit#9, startTime:8.723}**
Ciara	{Visit#106, startTime:3.0} {Visit#107, startTime:7.0} **{Visit#2, startTime:8.39}**
Anne	**{Visit#11, startTime:7.921} {Visit#10, startTime:8.633}**
Key	**{Visit#6, startTime:7.151} {Visit#14, startTime:8.847}**

Table 30 Final Schedules in Run 4

5.6 Summary

The designed AnyLogic-based simulation model aids health care agencies in analyzing their scheduling strategies through a low-cost simulation of the business process. The model simulates scheduling procedures following the proposed scheduling strategy, as well as practitioners' visiting processes. The preferences of both practitioner and patient can be imported into the model, including initial schedules, practitioner availability, and other preferences. The statistical simulation results are summarized in Table 32. The first five columns list configurations of different runs. The last three columns present the overall cost to practitioners, the payment by health care agencies, and the payment of using FCFS. It is observed that the payment of solutions generated in dynamic environments is reduced by an average of 38% compared to those generated by the FCFS algorithm across the eight groups of problem instances. It is also observed that the agency pays the practitioners on average 25% over the costs of taking a visit.

Group	Agent #	Visit #	Epsilon	Rate	New Visit #	Taken Visit #	Overall Cost	Payment	FCFS
Run #1	5	6	5	0.5	4	4	$50	$55	$120
Run #2	5	6	10	0.5	4	4	$50	$60	$120
Run #3	5	9	5	0.6	11	10	$152	$200	$300
Run #4	5	9	10	0.6	11	10	$152	$220	$300
Run #5	5	12	5	1	9	9	$151	$175	$270
Run #6	5	12	10	1	9	9	$151	$180	$270
Run #7	7	12	5	1.5	14	14	$201	$225	$420
Run #8	7	12	10	1.5	14	14	$201	$270	$420

Table 31 Summary of Simulation Results

Chapter 6 Conclusions and Future Work

This thesis investigates modeling and computational issues in developing solutions to dynamic decentralized scheduling problems in home health care. The issue of how to dynamically allocate practitioners to visits to ensure service quality while keeping costs to a minimum is an important one for policymakers. Compared to the classic centralized scheduling problem, one of the challenges of home health care resource scheduling is the lack of complete information in the distributed environment, in which the valuation information of each practitioner is unknown to anyone else. Another challenge is that the time dimension increases the complexity of the problem. Unexpected situations need to be accommodated within the time frame of an already active schedule. Based on our analysis of the problem, our main efforts address these issues in the design of the agent-based system and the dynamic scheduling simulation model.

In the agent-based system, a combined negotiation protocol and reverse combinatorial bidding framework is proposed as the communication method between agents. The Health Care Agency is set up to play the role of mediator in charge of organizing the negotiation process. The Practitioner Agents are self-interested agents and make decisions according to their own strategies. The decision-making process is implemented to help practitioners with the bidding process. The bidding mechanism facilitates multilateral negotiation between practitioners, with the objective of reducing home care service costs. The uniqueness of the proposed approach is that it uses an iterative bidding framework to integrate the exploration of best visiting schedules for the practitioners with the requirement of covering all planned visits, coordinating the behaviors of all self-interested parties in a highly decentralized resource allocation environment. The solution achieves a high efficiency of the optimal solution and the objective of cost containment in home health care, while still leaving a reasonable profit room for practitioners.

To deal with the dynamic challenge, we proposed a solution framework with an algorithm that repeatedly invoked a meta-algorithm, the bidding mechanism, according to a pre-defined scheduling strategy. The AnyLogic-based simulation model is developed to simulate the dynamic scheduling procedure as well as the business operations. The model takes initial schedules as input and simulates the accommodation of unexpected situations. The model was designed with an agent-based method combined with discrete time events, and the agents' local decision-making processes are reflected in the design of state chart. The parameters can be easily adjusted according to different scheduling strategies without touching the overall model structure. Different scenarios are described in order to illustrate the dynamic scheduling process, and the goals are achieved according to the computational results. Moreover, the model provides a decision-making support tool for health care agencies and serves as a test bed for different scheduling strategies.

One of our intended future research directions is to enhance the rescheduling algorithm to further reduce overall costs. Current rescheduling policies do not allow the practitioner to decommit to a previous contract. Assuming that decommitment is not an issue, such a policy can be adapted based on leveled-decommitment contracting [53], such that the practitioner is actively allowed to decommit to a previous contract by paying a penalty to other parties. The penalty designed in the mechanism is intended to provide a certain level of control to the system, rather than encouraging the contract parties to conduct their obligations and to prevent the contract parties from decommiting.

Designing a bidding language specifically for agent-based home health care scheduling is an interesting and potentially useful research direction. Based on the structure of requirement-based bidding language, a simple and sufficiently expressive bidding language could be designed for home health care scheduling, leveraging the domain specific scheduling problem structure. In addition, while the current solution is adequate for relatively small-scale scheduling problems, it does not provide the required responsiveness with large scale data. Given that the richness of the language allows users to express complementarities over their preferences, implementable approximation mechanisms could be able to trade off solution quality for a polynomial time guarantee.

References

[1] Canadian Institute for Health Information, "National health expenditure trends, 1975 to 2014," Canadian Institute for Health Information, 2014.

[2] EXECUTIVE OFFICE OF THE PRESIDENT OF THE UNITED STATES, "Trends in health care cost growth and the role of the affordable care act," EXECUTIVE OFFICE OF THE PRESIDENT OF THE UNITED STATES, 2013.

[3] E. Lanzarone, A. Matta and E. Sahin. Operations management applied to home care services: The problem of assigning human resources to patients. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions On 42(6),* pp. 1346-1363. 2012.

[4] E. Benzarti, E. Sahin and Y. Dallery. Operations management applied to home care services: Analysis of the districting problem. *Decis. Support Syst. 55(2),* pp. 587-598. 2013.

[5] J. A. Castillo-Salazar, D. Landa-Silva and R. Qu. Workforce scheduling and routing problems: Literature survey and computational study. *Annals of Operations Research* pp. 1-29. 2014.

[6] S. Yalçındağ, A. Matta and E. Šahin. Operator assignment and routing problems in home health care services. Presented at Automation Science and Engineering (CASE), 2012 IEEE International Conference On. 2012, .

[7] B. Cheang, H. Li, A. Lim and B. Rodrigues. Nurse rostering problems—a bibliographic survey. *Eur. J. Oper. Res. 151(3),* pp. 447-460. 2003.

[8] S. Bertels and T. Fahle. A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem. *Comput. Oper. Res. 33(10),* pp. 2866-2890. 2006.

[9] M. Mutingi and C. Mbohwa. Home healthcare worker scheduling: A group genetic algorithm approach. 2013.

[10] S. V. Begur, D. M. Miller and J. R. Weaver. An integrated spatial DSS for scheduling and routing home-health-care nurses. *Interfaces 27(4),* pp. 35-48. 1997.

[11] E. Cheng and J. L. Rich. A home health care routing and scheduling problem. 1998.

[12] S. Yalcindag, A. Matta and E. Sahin. Human resource scheduling and routing problem in home health care context: A literature review. Presented at Proceedings of 37th Conference on Opisto Research Applied to Health Services, Cardiff. 2011, .

[13] M. Gamst and T. S. Jensen. "A branch-and-price algorithm for the long-term home care scheduling problem," in *Operations Research Proceedings 2011*Anonymous 2012, .

[14] S. Nickel, M. Schröder and J. Steeg. Mid-term and short-term planning support for home health care services. *Eur. J. Oper. Res. 219(3),* pp. 574-587. 2012.

[15] K. Braekers, R. F. Hartl, S. N. Parragh and F. Tricoire. A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience. *Eur. J. Oper. Res. 248(2),* pp. 428-443. 2016.

[16] M. Mutingi and C. Mbohwa. Home healthcare staff scheduling: A clustering particle swarm optimization approach. 2014, .

[17] C. Akjiratikarl, P. Yenradee and P. R. Drake. PSO-based algorithm for home care worker scheduling in the UK. *Comput. Ind. Eng. 53(4),* pp. 559-583. 2007.

[18] M. S. Rasmussen, T. Justesen, A. Dohn and J. Larsen. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *Eur. J. Oper. Res. 219(3),* pp. 598-610. 2012.

[19] P. Eveborn, P. Flisberg and M. Rönnqvist. Laps care—an operational system for staff planning of home care. *Eur. J. Oper. Res. 171(3),* pp. 962-976. 2006.

[20] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger and G. R. Raidl. Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research 23(1),* pp. 89-113. 2013.

[21] L. A. Zadeh. Fuzzy sets. *Information and Control 8(3),* pp. 338-353. 1965.

[22] M. Mutingi and C. Mbohwa. A fuzzy simulated evolution algorithm for multi-objective homecare worker scheduling. Presented at Industrial Engineering and Engineering Management (IEEM), 2013 IEEE International Conference On. 2013, .

[23] M. Mutingi and C. Mbohwa. A novel simulated metamorphosis algorithm for homecare nurse scheduling. Presented at Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference On. 2014, .

[24] K. Thomsen. *Optimization on Home Care* 2006.

[25] P. M. Duque, M. Castro, K. Sörensen and P. Goos. Home care service planning. the case of landelijke thuiszorg. *Eur. J. Oper. Res. 243(1),* pp. 292-301. 2015.

[26] A. Trautsamwieser, M. Gronalt and P. Hirsch. Securing home health care in times of natural disasters. *OR Spectrum 33(3),* pp. 787-813. 2011.

[27] A. Trautsamwieser and P. Hirsch. A Branch-Price-and-Cut approach for solving the medium-term home health care planning problem. *Networks 64(3),* pp. 143-159. 2014.

[28] D. S. Mankowska, F. Meisel and C. Bierwirth. The home health care routing and scheduling problem with interdependent services. *Health Care Manag. Sci. 17(1),* pp. 15-30. 2014.

[29] Y. Kergosien, C. Lenté and J. Billaut. Home health care problem: An extended multiple traveling salesman problem. Presented at 4th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA'09), Dublin (Irlande). 2009, .

[30] R. Redjem, E. Marcon and X. Xie. Management of the care activities in home health care services: The routing and scheduling of caregivers level. Presented at Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference On. 2014, .

[31] M. Mutingi and C. Mbohwa. Home healthcare staff scheduling: A taxonomic state-of-the-art review. Presented at Industrial Engineering and Engineering Management (IEEM), 2013 IEEE International Conference On. 2013, .

[32] J. Bajo, J. A. Fraile, B. Pérez-Lancho and J. M. Corchado. The THOMAS architecture in home care scenarios: A case study. *Expert Syst. Appl. 37(5),* pp. 3986-3999. 2010.

[33] M. L. De Grano, D. Medeiros and D. Eitel. Accommodating individual preferences in nurse scheduling via auctions and optimization. *Health Care Manag. Sci. 12(3),* pp. 228-242. 2009.

[34] E. Kutanoglu and S. D. Wu. On combinatorial auction and lagrangean relaxation for distributed resource scheduling. *IIE Transactions 31(9),* pp. 813-826. 1999.

[35] A. Das and D. Grosu. Combinatorial auction-based protocols for resource allocation in grids. Presented at 19th IEEE International Parallel and Distributed Processing Symposium. 2005, .

[36] H. C. Lau, S. F. Cheng, T. Y. Leong, J. H. Park and Z. Zhao. Multi-period combinatorial auction mechanism for distributed resource allocation and scheduling. Presented at Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology. 2007, .

[37] F. Dargahi, M. F. Bhuiyan, C. Wang and C. Ding. Prioritizing and scheduling service requests under time constraints. Presented at Services Computing (SCC), 2014 IEEE International Conference On. 2014, .

[38] C. Wang and F. Dargahi. Service customization under capacity constraints: An auction-based model. *J. Intell. Manuf. 24(5),* pp. 1033-1045. 2013.

[39] H. Mehrizi and C. Wang. Iterative combinatorial auction for carrier collaboration in logistics services. Presented at 2013 IEEE International Conference on Systems, Man, and Cybernetics. 2013, .

[40] C. Xu, L. Song, Z. Han, Q. Zhao, X. Wang, X. Cheng and B. Jiao. Efficiency resource allocation for device-to-device underlay communication systems: A reverse iterative combinatorial auction based approach. *IEEE J. Select. Areas Commun. 31(9),* pp. 348-358. 2013.

[41] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput. (12),* pp. 1104-1113. 1980.

[42] T. Knabe, M. Schillo and K. Fischer. Improvements to the FIPA contract net protocol for performance increase and cascading applications. Presented at In International Workshop for Multi-Agent Interoperability at the German Conference on AI (KI-2002. 2002, .

[43] Protocol, FIPA Iterated Contract Net Interaction. Specification available online at www. fipa. org/specs/fipa00030. *XC00030F.Pdf* .

[44] G. Bullock, W. Nick, K. Asamene and A. C. Esterline. A multiagent and workflow system for structural health monitoring using the contract net protocol and alternatives. Presented at Maics. 2015, .

[45] E. Lanzarone, C. Masclet and F. Noel. Collaboration enhancement through tangible metaphors: Application to home care rescheduling. Presented at Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference On. 2013, .

[46] E. Lanzarone, C. Masclet and F. Noel. Qualitative assessment of a collaborative multi-user tool for enhancing the daily replanning and control of visits in home care services. Presented at Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference On. 2014, .

[47] P. Godin and C. Wang. Agent-based outpatient scheduling for diagnostic services. Presented at Systems Man and Cybernetics (SMC), 2010 IEEE International Conference On. 2010, .

[48] D. Gupta and B. Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions 40(9),* pp. 800-819. 2008.

[49] M. Guo, M. Wagner and C. West. Outpatient clinic scheduling: A simulation approach. Presented at Proceedings of the 36th Conference on Winter Simulation. 2004, .

[50] G. E. Vieira, J. W. Herrmann and E. Lin. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling 6(1),* pp. 39-62. 2003.

[51] D. Weyns, N. Boucké, T. Holvoet and B. Demarsin. DynCNET: A protocol for dynamic task assignment in multiagent systems. *Saso 7*pp. 281-284. 2007.

[52] W. Shen and D. H. Norrie. An agent-based approach for dynamic manufacturing scheduling. Presented at Proceedings of Workshop on Agent-Based Manufacturing. 1998, .

[53] T. Sandholm and V. Lesser. Leveled-commitment contracting: A backtracking instrument for multiagent systems. *AI Magazine 23(3),* pp. 89. 2002.

[54] M. Blais, S. D. Lapierre and G. Laporte. Solving a home-care districting problem in an urban setting. *J. Oper. Res. Soc. 54(11),* pp. 1141-1147. 2003.

[55] N. Lahrichi, S. Lapierre, A. Hertz, A. Talib and L. Bouvier. Analysis of a territorial approach to the delivery of nursing home care services based on historical data. *J. Med. Syst. 30(4),* pp. 283-291. 2006.

[56] D. Boldy and N. Howell. The geographical allocation of community care resources--A case study. *J. Oper. Res. Soc.* pp. 123-129. 1980.

[57] A. Hertz and N. Lahrichi. A patient assignment algorithm for home care services. *J. Oper. Res. Soc. 60(4),* pp. 481-495. 2009.

[58] A. Errarhout, S. Kharraja and I. Zorkani. Caregivers' assignment problem in home health care structures. Presented at Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference On. 2013, .

[59] C. R. Busby and M. W. Carter. A decision tool for negotiating home care funding levels in ontario. *Home Health Care Serv. Q. 25(3-4),* pp. 91-106. 2006.

[60] V. De Angelis. Planning home assistance for AIDS patients in the city of rome, italy. *Interfaces 28(3),* pp. 75-83. 1998.

[61] G. Merkuryeva and V. Bolshakovs. Vehicle schedule simulation with AnyLogic. Presented at Computer Modelling and Simulation (UKSim), 2010 12th International Conference On. 2010, .