aHome Healthcare Scheduling Engine

A Project Report

Submitted by

Harshavardhan M Sali (EE17B128)

in partial fulfillment of the requirements
for the award of the degree of

DUAL DEGREE
(BACHELOR AND MASTER OF TECHNOLOGY)

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.
30 MAY 2022

THESIS CERTIFICATE

This is to certify that the thesis titled **Home Healthcare Scheduling Engine**, submitted by **Harshavardhan M Sali(EE17B128)** to the Indian Institute of Technology, Madras, for the award of the degree of **Dual degree (B.Tech and M.Tech)**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Usha Mohan
Research Guide
Professor
Dept. of Management Studies
IIT-Madras, 600 036

Prof. Srirama Srinivas
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 30-05-2022

ACKNOWLEDGEMENTS

Firstly, I would like to express my immense gratitude to my advisor Prof. Usha Mohan, Department of Management Studies, for the continuous support and guidance throughout my Dual Degree project, and her patience, motivation, enthusiasm, and immense knowledge. I have gained much knowledge and insights about the subject through my interactions with her. Her flexibility, cooperativeness, and understanding have helped me complete this project. Prof. Usha provided me with great opportunities to learn and explore the field of operation research throughout the project duration. She showed me how to approach research difficulties, how to break them down and address each component at a fundamental level before progressing.
I would like to express my sincere and heartfelt gratitude to Prof. G Srinivasan and Prof. Rahul Marathe for their time and continued assistance and advice during the project.

My sincere thanks also go to Prof. Srirama Srinivas, Department of Electrical Engineering, my co-guide on this project and my faculty advisor, and for his time and continuous motivation not only for the project but also for my entire college duration. He made me believe in myself and inspired me during rough phases of college and helped me complete this project.

I would like to thank Nadkarni Omkar Madhav me19b196, B.tech student, Department of Mechanical Engineering, for his encouragement, insightful comments, discussions, and contribution to the project.

I would like to thank my parents and the institute faculty members for motivating me and being supportive throughout the work. I also take this opportunity to thank my friends who have helped and co-operated throughout the course of this project with stimulating discussions and conversations.

ABSTRACT

Human resource planning for home healthcare is a significant and difficult endeavor that involves a number of complicated elements such as labor regulation, caregiver preferences, and demand uncertainty. And the relevance of home healthcare is rapidly increasing as the population of industrialized and emerging nations ages and the number of hospitals, retirement homes, and medical personnel does not keep pace. In the problem under consideration in this paper, we attempt to assign caregivers to clients in the most efficient manner possible, which is currently done manually by most home healthcare agencies, which results in longer planning times and suboptimal decisions that frequently fail to meet the population's health needs. We propose a two-stage optimum programming paradigm, using Gale Shapley Algorithm for assignment and Linear programming for scheduling in-home healthcare. We compare our approach between Gurobi Optimizer and PuLP python library for linear programming problems.

KEYWORDS: Home healthcare · Optimisation · Heuristics · Simulation · Scheduling · Linear Programming · Gale Shapley · Gurobi · PuLP ·

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
OR	Operations Research
TSP	Traveling Salesman Problem
HHC	Home Healthcare
HC	Healthcare
GDP	Gross domestic product
MILP	Mixed-integer linear programming
MIQP	Mixed-integer quadratic programming
MIQCP	Mixed-integer quadratically-constrained programming
QP	Quadratic programming
PuLP	Python library for linear Programming

NOTATIONS

d: 1-7 Days (Mon,Tue,..,Sun)

i,j: Clients

u_{id}: Variable to stop subtours i.e once a caregiver visits client 'i' the variable ensures that caregiver doesn't return in client 'i'

a: Number of clients

VAR_{ijd}: Caregiver goes from 'i' to 'j' on day 'd'

t_{Yxij}: 't' travel time matrix

Y: Discipline/skill

x: Caregiver ID

z_{jd}: Number of hours of service a client 'j' receives on day 'd'

VAR_{0jd}: Caregivers go from '0' his house to a particular client 'j' on day 'd'

$VAR_{i(a+1)d}$: Caregivers go from 'particular i' client to his house 'a+1' on day 'd'

Cg: Set of caregivers

Cl: Set of clients

CHAPTER 1

INTRODUCTION

Home healthcare (HHC), also known as in-house care, social care, or domiciliary care, is any sort of treatment provided to a patient at their home rather than in a healthcare institution such as a hospital or clinic. HHC is typically given to the elderly and terminally ill in order to preserve or improve their quality of life. Other patient groups, such as children, post-surgical patients, and so on, may benefit from HC services. The fundamental advantage of Home Care is that it reduces hospitalization rates, which lowers overall healthcare costs[1].

1.1 What services can be offered in Home Settings

The HHC market consists of products and services. HHC services must be adaptable and tailored to meet the requirements of each individual. The following are some of the services:

- Health education: nutrition, fall prevention, healthy lifestyles, and other topics for the elderly and their family members.
- Personal hygiene: exercising and monitoring vital indicators including blood pressure, pulse, heart rate, and blood glucose level.
- Early identification and prevention: avoidance of bed ulcers, healing of wounds as needed, blood pressure measurement, frequent laboratory testing, and breast self-examination.
- Psychosocial assistance and social services, including counseling for seniors and family members.
- Building the capacity of family members to provide day-to-day care.
- Management of simple diseases and follow-up

1.2 Why focus on Home Healthcare problems?

HHC's most important goal is to guarantee that those who require medical treatment and everyday care receive high-quality home care. The population aging, as well as the fast rise in the proportion and quantity of older adults, is a global phenomenon caused by lower fertility and longer life expectancies[2]. More individuals than ever before are predicted to be in danger of experiencing chronic illnesses, bad health, and dependency that commonly accompany late age by 2030. According to estimates, there will be two billion individuals over 60 years old in 2050, with 80 percent of them residing in developing nations[3].

1.3 Home Healthcare in the US vs. India

Both the national and state governments run India's universal, decentralized healthcare system. Medical education is overseen by the national government, which also gathers information on infectious illnesses. Although attempts are being made, the United States does not yet have a universal health care system.

India spends roughly $40 per person on health care each year, compared to $8,500 in the United States. India's GDP is $1.6 trillion, whereas the United States' healthcare spending is $2.6 trillion. The United States spends over 16 percent of its GDP on health care, which is more than the global average. Only a small fraction of the Indian population has access to medical insurance. In India, the general people have virtually little knowledge about it or its advantages. The amounts paid out by accessible insurance policies are out of date and do not represent current healthcare costs. As a result, the majority of Indian doctors favor uninsured patients. Medical insurance is a critical component of the United States healthcare system.

Even without a doctor's prescription, one may readily obtain drugs over the counter in India. Sometimes one may relate to the pharmacist's condition and is given medication as a result. This would not be possible in the United States.

India continues to be hampered by a dismal doctor-to-patient ratio of one doctor to 1,674 people. Hospitals are usually always packed, with huge lines at every counter, from billing counters to investigation rooms. Home healthcare, with its advantages and hassle-free approach, is a silver lining in the healthcare industry at this moment. Doctors and caregivers would come to the house to treat patients. Home healthcare also fits the bill wonderfully when it comes to taking care of the health of all family members, particularly ailing and infirmed family members.

India's requirement for medical attention and services will continue to expand as the world's second-most populous country. Considering these factors, it's reasonable to predict that the need for home healthcare will grow in the next few years. In the next thirty years, it is expected that the number of elderly people will grow thrice. More in-person medical attention and care will be required than is already required. Furthermore, the growing number of individuals suffering from chronic conditions like cancer, renal failure, Alzheimer's disease, and other ailments would greatly benefit from this module of healthcare service and will be able to live a more self-reliant, independent, and dignified life as a result.

1.4 Effect of Covid-19 on Home Healthcare

The impact of the COVID-19 pandemic on the home healthcare business is ambiguous. Because the home healthcare industry comprises both products and services, each has a different impact. COVID-19 would have a good effect on a variety of home healthcare monitoring products, including blood glucose monitors, blood pressure monitors, pulse oximeters, and temperature monitors.

Since the COVID-19 pandemic, one may assume that the home healthcare sector would thrive because providers would be able to treat the weak and older adults in the comfort of their own homes. The situation, however, is not the same. Many home healthcare companies and organizations have experienced a drop in business.

1.5 Problems faced by Home Healthcare Agencies

There are several issues that home healthcare providers must deal with, including:

1. Patient preference. As the usage and acceptability of home-based care develops, it is critical to examine patient preferences for home-based care vs. traditional brick-and-mortar care.
2. Clinicians' concerns: Caring for patients at home necessitates lengthier visits than in a hospital or office setting, resulting in smaller panel size.
3. Infrastructure: It's difficult to manage patients' acute care demands at home due to a lack of supporting infrastructure.
4. Human resources shortage to meet the demand of healthcare requests

This project mainly focuses on improving HHC from a product point of view to optimally manage human resources with assignment and scheduling of caregivers so as to meet the demand of the services requested by the clients. The industry practice with most home healthcare agencies so far has been to manually schedule patient visits for agency workers with certain thumb rules for efficient utilization of the workforce, which is a time-consuming and complex process. The objective of this paper is to automate the complex task of scheduling caregivers for patients so that healthcare providers can maximize their patient care activities while reducing the required effort and costs by ensuring optimized allocation of patient care activities to their scarce resources. In particular, the model takes into account patient's and caregiver preferences, as well as caregiver's availability and skills, and gives an optimal short-term (daily/weekly) schedule.

In the following sections, we will conduct a literature review of previous research on hospital healthcare planning challenges. Following that, we describe our problem statement, the methods we recommend for effectively addressing the problem, and ultimately, the outcomes and conclusion.

CHAPTER 2

LITERATURE REVIEW

Over the last few years, a lot of research has been done on hospital healthcare planning issues. Nurse staffing and scheduling issues have gotten the greatest attention from the operations research (OR) community because producing high-quality nurse schedules may enhance hospital resource efficiency, patient safety and satisfaction, and administrative workload [4].

Tables 2.1, 2.2, and 2.3 represent a classification of publications in terms of the planning horizon, uncertainty, objectives/performance measures, constraints, and solution methodologies.

2.1 Classification based on performance measure and objectives

Table 1: A classification of publications in terms of performance measure and objective

	Time Travel/Cost	Wait Time/Cost	Resource Utilization	Patient/Staff Preference	Unscheduled Patient/Task
Begur et al.	✓	✓			
Gaspero and Urli	✓				✓
Bard et al.	✓				
Carello et al.	✓				
Cappanera et al.	✓	✓			
Duque et al.	✓	✓		✓	
Zhan et al.	✓				
Hiermann et al.	✓	✓		✓	
Brackers et al.	✓			✓	
Bennett and Erera					
Mankowska et al.	✓	✓			✓
Our study	✓	✓	✓	✓	✓

Travel time/cost, times that workers spend between visits and cost is caused by different transportation; Waiting time/cost, times that workers spend by waiting if they arrive location of patients earlier than starting times; Patient/staff preferences, satisfaction level of preferences and desires of patients and workers; Unscheduled patient/task, performing as many tasks or accepting as many patients as possible

2.2 Classification in terms of constraints

Table 2: A classification of publications in terms of constraints

	Skill Matching	Multi Worker	Time Window	Consistency/Periodicity	Patient/Staff Preference
Begur et al.	✓		✓	✓	
Gaspero and Urli			✓		
Bard et al.			✓		✓
Carello et al.			✓	✓	
Cappanera et al.	✓		✓	✓	
Duque et al.	✓			✓	
Zhan et al.			✓		
Hiermann et al.	✓		✓		✓
Braekers et al.			✓		
Bennett and Erera				✓	
Mankowska et al.	✓	✓	✓		
Our study	✓	✓	✓		✓

Skill matching, qualification of nurses must be sufficient for needs of patients; Multi worker, more than one worker must simultaneously perform the same task; Time windows, time intervals that visits must be performed in; Consistency/periodicity, patients must be serviced by same times and nurses during their service horizons; Breaks, rest times for nurses after working prespecified hours.

2.3 Classification in terms of solution methodologies

Table 3: A classification of publications in terms of solution methodologies

	Exact	Heuristic	Single objective	Multi Objective	Static	Dynamic
Begur et al.		✓	✓		✓	
Gaspero and Urli		✓	✓		✓	
Bard et al.	✓	✓	✓		✓	
Carello et al.	✓		✓		✓	
Cappanera et al.	✓		✓		✓	
Duque et al.		✓		✓	✓	
Zhan et al.	✓	✓	✓		✓	
Hiermann et al.		✓	✓		✓	
Braekers et al.		✓		✓	✓	
Bennett and Erera		✓	✓		✓	
Mankowska et al.	✓	✓	✓			✓
Our study	✓	✓		✓	✓	

Static problem settings, all data are known in advance before the optimization has started; Dynamic problem settings, new patients are arrived as time progresses and must be dynamically incorporated into an evolving schedule

2.4 Summarization of previous research works

Valeria Borsani and Andrea Matta propose a linear integer scheduling model that may be used to facilitate short-term human resource planning in-home care. The approach they propose addresses the issue of determining (a) which human resource to utilize and (b) when to execute the service throughout the planning horizon in order to meet the care plan for each patient treated by Home Care providers[6]. The solution is hierarchical and consists of two linear programming

models: the first concerns the assignment of new patients to a reference operator, and the second concerns a scheduling model, the output of which is the weekly plan for each operator.
Using this paradigm during the planning process can provide several benefits. First and foremost, the service quality and, as a result, the patient's pleasure may be increased. Furthermore, nurse coordinators may be solely responsible for supervising the procedure and dealing with unexpected situations. A third advantage might assist providers that operate in several districts: they could use a single central planning system to better use human resources on the ground than local sub-optimal planning.

Redjem and Marcon[7] offer an HHC service challenge that includes several visits to the same patient in a day as well as time dependencies based on ranking tasks in terms of their relationships. They want to cut down on caregivers' waiting and travel time while they're working within a tight schedule. They propose a two-stage heuristic for caregiver routing: The shortest trip time for each caregiver is determined in the first stage, with no patient coordination of sequencing constraints. The final answer is merged with all assumptions and limitations in the second stage.

Issabakhsh et al.[8] offer a solid mathematical model for peritoneal dialysis patients at home. Patients, according to their model, have a variety of needs, including the collection of urine or blood samples, visits from nurses and technicians, and the delivery of certain medications. They must consider not just depots and patient locations but also dialysis centers and laboratories because of these requirements. They also take into account various limitations, such as the need to visit labs after collecting blood or urine samples and the need to pick up nurses and technicians from dialysis centers before visiting patients. They create a robust optimization model to accommodate variability in journey times because just-in-time visits are a critical factor for peritoneal dialysis.

Guericke and Suhl[9] construct an HHC model that takes work regulations and legal considerations into account. They consider break-rest intervals, weekly work lengths, and shift rotations in accordance with German rules and regulations in order to examine their impact on outcomes. For a small-size issue, they present a mixed-integer linear solution. Furthermore, an adaptive big neighborhood search-based heuristic is included to deal with a real-size difficult problem in a fair amount of time. In comparison to the mixed-integer program, the heuristic technique performs better in terms of execution time.

Frifita et al.[10] provide a model for an HHC problem, including time frames and synchronization, in which numerous carers visit a patient at the same time. They present a broad variable neighborhood search strategy for reducing caregiver trip time. For a range of real-life scenarios, the proposed methodology is contrasted against a mixed-integer programming model and a heuristic strategy. When compared to the mixed-integer programming model, their method is faster and produces results that are close to the ideal solution.

Wirnitzer et al.[11] create a monthly nurse rostering model for an HHC firm in order to automate previously manual scheduling processes. They suggest five formulations for mixed-integer programming. Each has a particular goal function aimed at ensuring continuity of care for patients while adhering to the same strict limits, such as breaks, maximum daily and weekly working durations, patient/nurse preferences, and shift rotations. All models outperform human planning intolerable time, according to results based on randomly generated data drawn from real-world input and the company's data. They also compare the results of the models in terms of computational times, the number of assigned and swapped nurses, and other factors.

Existing papers in the literature generally focus on static problem settings for which all patient requests are fully known, but very few focus on optimal multi-objective methodologies accounting multi workers, patients/caregivers' preferences, and qualification matching. Given the literature gap, the proposed model aims to provide a well-rounded 2-phased automated scheduling engine that will handle a set of individual patient requests that need to be scheduled to caregivers in a particular time window. Once the engine receives a list of individual patient visits enumerated from the Agency's multiple visit authorizations, by default, all visits are considered for scheduling, but the user has the option to exclude visits or define a subset of visits to be scheduled. When scheduling several branches of an agency, the geography of the branches will be taken into account. This means that geographically separated branches will be planned individually, whilst co-located branches may be scheduled jointly to take advantage of their closeness. The scheduling engine guarantees that numerous hard and soft restrictions are met while determining the lowest distance solution.

CHAPTER 3

HOME HEALTHCARE SERVICE

3.1 Problem Statement

The problem we consider is the multiple caregiver/nurses HHC assigning and scheduling problem in a static environment. We solve the problem optimally using Gurobi Optimiser and heuristically by using PuLP linear programming.

Bertels and Fahle [12] presented a model to solve the assignment and scheduling problems at the same time, comparing different mathematical solutions to this problem; Eveborn et al. [13] dealt only with scheduling, considering the care continuity as a constraint and using a repeated matching method; Begur et al. [14] developed optimization models for scheduling through Integer Mixed Programming and, above all, for routing by comparing some heuristic procedures.

The solution presented allows for to adaptation of technical methods used in industry to the HHC service. A hierarchical structure composed of assignment and scheduling models similar to the one used in industrial planning is proposed.

3.1.1 Clients/Patients

Each patient/client can request a specific service for a specific time period, such as a week, month, or year. In a specific time period, each client can make repeated requests for different or identical services. Tokens (Authorization IDs) will be generated for each request made by the clients.
HHC agencies must be paid in advance for the services that clients seek to claim in the industry. Clients' payments are considered as units, indicating the number of units handled by caregivers/nurses throughout the time period chosen by the clients.

A request from **i** for a service/discipline of $\mathbf{d_i}$ from location $\mathbf{l_i}$ for the time interval $\mathbf{t_i}$ will generate authorization ID, $\mathbf{a_i}$ with their total units paid for, $\mathbf{ut_i}$ and total units left in their claim, $\mathbf{u_i}$.

3.1.2 Caregivers/Nurses

Each caregiver specializes in a certain discipline/skill, such as pediatric therapy, pediatric nursing, physician house calls, physical therapy, and so on. As a result, each caregiver should be correctly mapped to clients who want their services.

There are labor standards that the sector must observe when it comes to employment; we also examine the availability of carers. Because of holidays, meetings, training courses, protracted illness, or personal reasons, a caregiver may be absent. We can't overwork caregivers; therefore, we set a maximum limit of 8 hours per day per caregiver. In order to optimize the scheduling engine, the approach we propose also considers minimizing the distance/time traveled between caregivers and clients.
All the clients are visited by caregivers in the schedule order generated by our scheduling system.

3.1.3 General assumptions

- All authorized patient visits are completed
- All pinned visits are retained as planned
- Each planned visit is scheduled only once during the week
- Each visit is assigned before its authorization expiry date
- Only one visit of a specific type is scheduled in a day for each patient
- Patient’s preferred days of visit are accommodated to the optimal extent
- Total number of visits in each day requiring a specific skill does not exceed the resources available
- Once the caregiver leaves home to serve the client, he completes all of his scheduled visits before returning home

- Unforeseen events, such as operators' sudden unavailability or changes in patient's conditions, are not considered

3.2 Dataset

Data was simulated after engaging with an end-to-end home healthcare provider to apply industry-standard technological approaches to the scheduling engine.

We have simulated client requests, and time frames at random. Where as client discipline requests and caregivers were generated in such a way that we could cover all industrial scenarios, such as when client requests and caregivers are almost equal and when client requests are far larger than caregivers accessible to service the clients, and certain edges where we have caregivers but no client demands for that discipline.

3.2.1 Client dataset

Simulated data contains **748** distinct clients with **1134** permission requests in total, with randomly generated time frames and disciplines from when they wish to claim the service until when, and a weekly restriction of merely the number of units the client wants serviced each week.

We have simulated **6** unique disciplines, SN, ST, PT, CNA, MSW, OT with randomized allocation of client requests of these disciplines.

Table 4 Client authorization details

CLIENT_ID	AUTH_ID	DISCIPLINE	FROM_DATE	TO_DATE	TOT_UNITS	UNUSED	WeeklyLimit
1	1280	OT	2021-12-04 00:00:00.000	2022-06-30 00:00:00.000	0	16	0
2	2152	SN	2021-11-07 00:00:00.000	2022-11-06 00:00:00.000	3120	2997.07	60
3	2052	SN	2021-09-18 00:00:00.000	2022-09-17 00:00:00.000	6136	6136	168
3	96	CNA	2021-02-15 00:00:00.000	2022-02-14 00:00:00.000	7300	1201	39

3	1144	CNA	2022-02-15 00:00:00.000	2023-02-14 00:00:00.000	7300	7300	140
4	784	SN	2021-09-28 00:00:00.000	2022-09-27 00:00:00.000	8736	8541.58	168
5	428	CNA	2021-02-28 00:00:00.000	2022-02-27 00:00:00.000	4745	55	91
5	1632	CNA	2022-02-28 00:00:00.000	2023-02-27 00:00:00.000	4745	4563	91
6	1814	PT	2021-04-26 00:00:00.000	2022-04-25 00:00:00.000	0	44	0
6	440	OT	2021-04-26 00:00:00.000	2022-04-25 00:00:00.000	0	97	0
6	1796	ST	2021-05-13 00:00:00.000	2022-05-12 00:00:00.000	0	70	0
7	46	ST	2021-05-26 00:00:00.000	2022-05-25 00:00:00.000	0	36	0
7	952	PT	2021-06-18 00:00:00.000	2022-06-17 00:00:00.000	0	18	0
7	860	OT	2021-06-27 00:00:00.000	2022-06-26 00:00:00.000	0	87	0
7	70	CNA	2022-01-04 00:00:00.000	2023-01-03 00:00:00.000	4015	3246	77
8	1638	OT	2021-09-22 00:00:00.000	2022-06-30 00:00:00.000	0	34	0
8	1630	ST	2021-09-22 00:00:00.000	2022-06-30 00:00:00.000	0	43	0
9	1522	ST	2021-09-22 00:00:00.000	2022-02-06 00:00:00.000	0	0	0
9	818	ST	2022-02-07 00:00:00.000	2023-02-06 00:00:00.000	0	48	0
10	1002	ST	2021-09-22 00:00:00.000	2022-02-08 00:00:00.000	0	0	0
10	884	ST	2022-02-09 00:00:00.000	2023-02-08 00:00:00.000	0	46	0
11	1348	OT	2022-01-06 00:00:00.000	2023-01-05 00:00:00.000	0	50	0
12	1656	PT	2021-08-06 00:00:00.000	2022-02-06 00:00:00.000	0	0	0
12	1014	OT	2021-10-06 00:00:00.000	2022-06-30 00:00:00.000	0	59	0
12	1670	PT	2022-02-07 00:00:00.000	2023-02-06 00:00:00.000	0	94	0

Tot_units are the total number of units the client has while unused gives the total number of ***Unused*** units the client has left to claim services. The ***WeeklyLimit*** is a weekly restriction of merely the number of units the client wants serviced each week.

Table above shows possible ways clients can request for services,

1. Each client can request for various services either same or different in the same or overlapping time window

   Eg. Client 12 has requested for PT, OT and PT services.
2. Client can request for only one service in a given time window

   Eg. Client 1, 11

Each authorization ID is unique because a single client might request various services or the same services several times within the defined time window.

Table 5 Client location details

CLIENT_ID	Latitude	Longitude
1	39.422356999999998	-104.88208450000001
2	38.294862000000002	-104.6194105
3	39.383821699999999	-104.9067876
4	39.993233400000001	-104.7744051
5	39.605700499999998	-104.7476959
6	39.676025099999997	-104.917491
7	39.649393199999999	-105.0309699
8	39.758094999999997	-104.8317359
9	39.823298999999999	-105.0535041
10	39.823298999999999	-105.0535041
11	39.890247899999999	-104.7998879
12	39.936281299999997	-104.9407562
13	39.873938299999999	-104.94158330000001
14	39.802537200000003	-104.76768610000001
15	38.7305128	-104.6902822
16	39.714911499999999	-104.6856126
17	39.971687099999997	-104.8130958
18	39.685029200000002	-104.8225023
19	38.978581200000001	-104.74632819999999
20	39.742446200000003	-104.9050084
21	39.826641000000002	-105.0171833

22	38.8139684	-104.7714422
23	39.640439399999998	-104.8223644
24	39.992059300000001	-105.0633674
25	39.659235700000004	-104.8615083

Table 5 illustrates how the geolocation of a client has been recorded in terms of latitude and longitude.

3.2.2 Caregivers dataset

Simulated data includes **107** caregivers at various places (latitude, longitude) to serve clients.

Table 6 Caregiver details

CAREGIVER_ID	Latitude	Longitude	BaseDisc_Or_AuthDisc
1	39.4043773	-104.8882145	SN
2	38.341721399999997	-104.715979	SN
3	38.930562999999999	-104.672168	SN
4	38.992553600000001	-104.7648753	SN
5	38.705290699999999	-104.8406052	SN
6	39.5315218	-104.9933544	OT
7	38.314374399999998	-104.7412928	SN
8	38.976653300000002	-104.5939127	SN
9	38.316170999999997	-104.62830700000001	SN
10	38.7435294	-104.6314527	SN
11	38.859769399999998	-104.76310100000001	CNA
12	39.7617549	-105.0356159	ST
13	38.827167299999999	-104.7499035	CNA
14	38.722110899999997	-104.65930950000001	SN
15	39.895663900000002	-104.85364800000001	SN
16	38.7268483	-104.6691153	CNA
17	38.990347999999997	-104.723311	SN
18	38.990347999999997	-104.723311	SN
19	39.658933900000001	-104.9897484	PT
20	38.228139599999999	-104.6226436	CNA
21	38.870245300000001	-104.6942158	SN
22	38.422886800000001	-105.2158927	SN
23	39.632655700000001	-104.9105849	ST
24	33.213317699999998	-94.691872799999999	SN
25	38.971518199999998	-104.71681169999999	CNA

CHAPTER 4

METHODOLOGY

In this piece, we will discuss our two-stage solution, Assignment, and Scheduling, which we developed with the Gurobi Optimizer and the PuLP linear programming library. Two greedy heuristics will also be discussed.

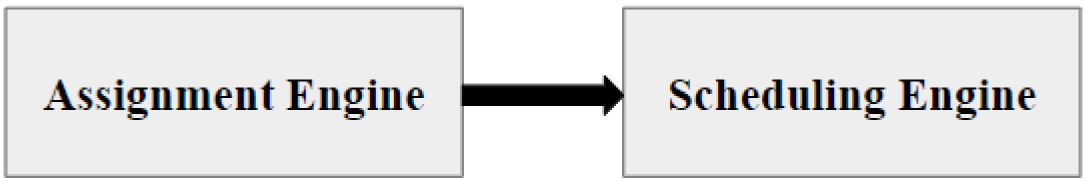

Fig 1 Scheduling Engine Stages

4.1 Stage 1: Assignment Engine

The assignment model is a preliminary optimization stage in which we try to create an optimized possible list of caregivers who can service each client, as well as a possible list of clients for each caregiver. We give priorities first to caregiver availability, then discipline/skill, and finally distance.

4.1.1 Gale-Shapley Algorithm

Pattern matching is a topic covered by several algorithms. The stable matching algorithm is a method for locating solutions to problems involving stable positions. Using this approach, a bipartite network will be produced that indicates matching between stable and optimum elements, which can be ideal on either the man or woman side.

In the Stable Marriage Problem, for example, both men and women are applicants, with a third kind serving as the application's recipient. For best measurement, the type of applicant group will be appraised. If a man serves as an applicant, the stable pair created will be optimum in comparison to other males; this also holds true if the applicant's position changes. The male role will be used as an applicant so that the results of this algorithm will produce optimal stable pairs relative to men.

One-To-One Matching: Stable Marriage Problem

The Stable Marriage Problem was presented by David Gale and Lloyd Shapley in 1962 as matching research to distribute a set of partners with stability. The objective is to discover a stable pair of X and Y instances. Each variable has its own matched pair preference list. Gale and Shapley's seminar paper "College Admissions and the Stability of Marriage" [16] was the first to mention the Stable Marriage Problem. Gale and Shapley developed the Gale-Shapley Algorithm to couple certain n things X with n objects Y using particular rules to solve the Stable Marriage Problem.

The first object is defined as n men, and the second object is n women in an arranged marriage, in which both parties have a preference list against each of the opposite sex. Gale-Shapley has a proposed rule for each man and each woman. In the process of algorithm execution, each man has an alternative pair and is free, but every woman must pair. Women are definitely in pairs even if their partners can change. Men who couple more than one get a couple who has the least preference for themselves. When a woman receives a proposal directly will be accepted and become a temporary partner. When a woman who has been in pairs receives a different application, she will compare it with the previous application and reject the man who has a smaller criteria fit against the female preference. Each man is applying to women 60 according to the criteria in order until later in pairs. If the application is rejected by one of the women on the list, then he is free again and continues the application sequence on his list. The algorithm ends when everything has been paired.

Example: Male's = {A,B,C} and Female's = {X,Y,Z}

A	B	C
X	X	Y
Y	Z	Z
Z	Y	X

X	Y	Z
B	B	C
A	C	A
C	A	B

Fig 2 Preference order of Males and Females

Let's start with the men as a proposing group.

I Iteration

A proposes to X, his first choice, and she agrees because X is now unemployed.

B proposes to X, his first choice, and because B is higher on X's list of priorities, she refuses A's offer and accepts B's.

As a result, A is currently without a companion.

C proposes to Y, his first choice, and Y accepts his offer because Y has no other proposals at the moment.

II iteration

B and C are already engaged, while A is still looking for a companion.

A proposes to Y, his second option, but she rejects him since she has a better offer (C).

We don't bother B and C because they are happily engaged.

III Iteration

A proposes to Z, his third and final choice, and she agrees since she has no other options.

We stop iterating after all of the males are engaged with their respective partners.

The Matching is: **{A Z},{B X},{C Y}.**

Let us now reverse the roles and make the females the ones proposing.

I iteration

X proposes to B, her first choice, and while B is now unemployed, he accepts her offer.

Y proposes to B, her first option, but B rejects since he already has a better offer (X). As a result, Y is without a partner.

Z proposes to C, her first option, and C agrees because he has no other alternatives.

II iteration

X and Z are engaged, but Y is looking for a partner.

C, Y's second option, receives a proposal from Y. Because Y is higher on C's priority list, he accepts Y's offer and refuses the offer from Z that he had previously accepted. As a result, Z is currently without a companion.

A is Z's second choice, and she proposes to her.

Because A has no other offers, he takes hers.

We cease iterating now that all of the women are engaged to their respective spouses.

The matching is: **{X B},{Y C},{Z B}.**

This match is identical to the one found when males were the proposing group.

Gale-Shapley Algorithm Pseudocode:

```
function stableMatching {
    Initialize all m ∈ M and w ∈ W to free
    while ∃ free man m who still has a woman w to propose to {
       w = first woman on m’s list to whom m has not yet proposed
       if w is free
         (m, w) become engaged
       else some pair (m', w) already exists
         if w prefers m to m'
            m' becomes free
           (m, w) become engaged
         else
           (m', w) remain engaged
    }
}
```

Fig 3 Gale-Shapley algorithm Pseudocode

Many-To-One Matching:

This is an extension of one-to-one Gale Shapley stable matching. Unlike one-to-one stable marriage matching, which requires an equal number of men and women, we do not require an equal number of men and women for many-to-one matching.

For our problem, we want to generate an optimized list of caregivers who can service each client, as well as a reference list of clients for each caregiver, i.e., many clients need to be matched with potential caregivers and vice versa. Finally, we use many-to-one Gale Shapley Matching to arrive at an optimum matching list rather than a random one.

Application of Many-To-One Gale Shapley Matching:

1. College admission
2. Mess allocation
3. Hospital Resident
4. Course allocation

```
assign each resident to be free

assign each hospital to be totally unsubsidised;

while (some hospital h is unsubsidised) and (h's list contains a
resident r not provisionally assigned to h) do

begin

r:= first such resident on h's list;

if r is already assigned, say to h', then

break the provisional assignment of r to h':

provisionally assign r to h;

for each successor h' of h on r's list do

remove h' and r from each other's lists

end;
```

Fig 4 Gale-Shapley (Many-To-One) algorithm Pseudocode

Example: The College admission problem

College choices typically impact students' futures and having a student-college match that optimizes the welfare of everyone involved is desirable. We can see, though, how difficult it is to come to this awareness. Colleges want to fill their quotas; therefore, they'll admit students who are more likely to accept their offers. Students' preferences are not widely known by a centralized authority that does the matching; thus, having a public preference list in this situation may actually hinder the student's chances of being admitted to a school lower on their list. The waitlist method, which has been in use for decades, places the highest-ranked students outside of the admitted quota on a waiting list and is admitted later if a space occurs as a result of other students refusing their offer. When a student is on the waitlist for a school that they like more than the one to which they have been admitted, they might lie about the waitlist and rescind their admission once they have been accepted by the first school.

The objective is to create a system that matches students and colleges in such a way that no two of them would choose each other over their existing pairings. With the stable marriage problem, we've encountered this concept of stable matching in class. The distinction is that each institution has the ability to admit more than one student.

The process works in a similar way to the postponed acceptance algorithm or stable marriage:

1. Every student submits an application to their top-ranked universities.
2. Each college with a quota q selects the q highest-ranking students from the applicants and places them on a waiting list while rejecting the remainder.
3. Students that are denied apply to their second-best institutions.
4. The institution evaluates the top-ranked students among those who applied and are on the waitlist and adds them to the list while rejecting the others.
5. The procedure is repeated until all students have been matched or have applied to all of the schools to which they wish to apply.

Only students who cannot be accepted to a college under any stable matching are rejected by the system, resulting in optimum matching.

4.1.2 Flowchart

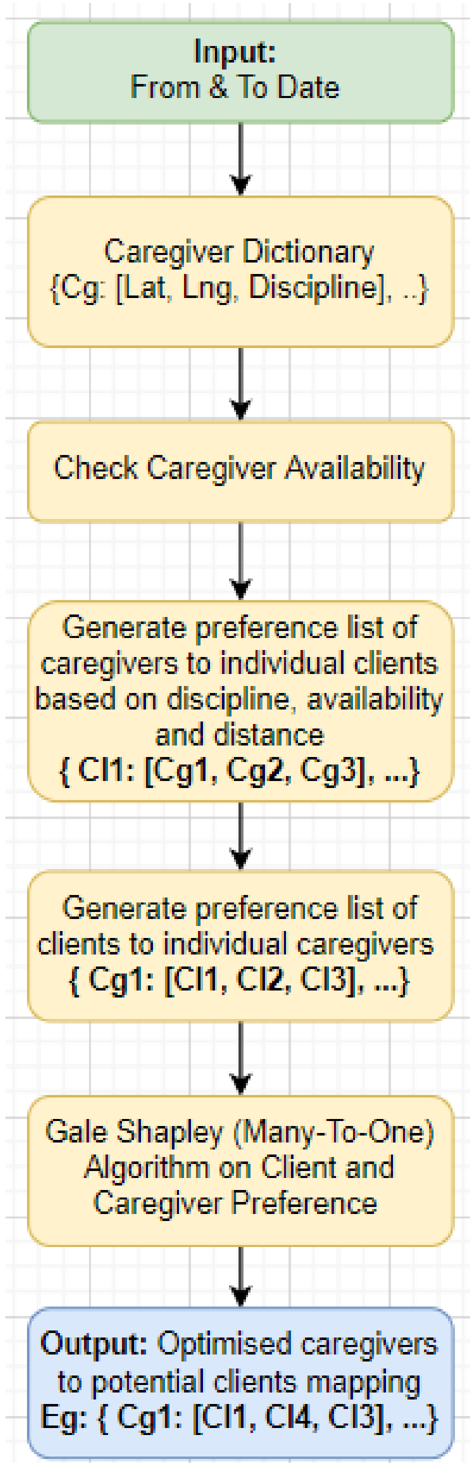

Fig 5 Assignment algorithm flowchart

Algorithm: We invoke the assignment stage before the scheduling algorithm to generate an optimized list of caregivers who can service each client, as well as a preference list of clients for each caregiver to finally use both the lists to generate an optimal list of potential clients to caregivers, i.e., $\mathbf{Cg_N = [Cl_1, Cl_2, Cl_3 \ldots Cl_M]}$ where Cg are caregivers and Cl are clients.

The goal is to build personalized schedules for each caregiver in a certain time window so that they may provide services to the clients. We accomplish this by:

1. Inputting the date, from, and to which we are trying to schedule caregivers for clients with active requests in that time range.

2. We create a python dictionary with all of the caregiver's information and check for their availability throughout that time frame.
   We do not evaluate a caregiver further if he/she has requested leave or is on vacation during that time period.

3. We generate a list of suitable caregivers for each client based on location, availability, and expertise of the caregiver after considering the client's preference for what discipline service they are wanting.

   Example:

   {

   Cl_1: [Cg_1, Cg_2, Cg_3, …],

   Cl_2: [Cg_5, Cg_3, Cg_7, …],

   .

   .

   Cl_M:[Cg_1, Cg_6, Cg_9, …]

   }

   We sort the list of caregivers by distance after prioritizing caregiver availability, implying that customers choose caregivers who have the expertise they requested and are close by.

4. We generate a similar list of clients to each caregiver and sort it based on the distance.

   Example:

   {

   Cg_1: [Cl_7, Cl_2, Cl_3, …],

   Cg_2: [Cl_{10}, Cl_{35}, Cl_{76}, …],

   .

   .

   Cg_N:[Cl_1, Cl_{61}, Cl_{99}, …]

   }

   Note: The number of clients and caregivers are not the same, as in the HHC industry, the number of clients and caregivers are not equal. Clients and requests constantly outnumber caregivers.

5. To build a more optimized list of clients for each caregiver, we use the many-to-one Gale Shapley algorithm to the client and caregiver preference lists generated in the preceding phases.

   Example:

   {

   Cg_1: [Cl_{10}, Cl_{67}, Cl_{31}, …],

   Cg_2: [Cl_{10}, Cl_{35}, Cl_6, …],

   .

   .

   Cg_N:[Cl_{11}, Cl_1, Cl_{96}, …]

   }

   We construct two sets of preferences, one from the perspective of the client and the other from the perspective of the caregiver because one client may have one caregiver at the top of their preference list but that does not mean that caregiver would have that client at the top of theirs. To negate this randomness and to achieve an ideal prospective list of clients that a caregiver can serve, we employ two sets of preferences and apply a many-to-one Gale Shapley algorithm.

Haversine formula: We use the Haversine formula to calculate the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation. The haversine can be expressed in trigonometric function as:

Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R \cdot c$$

where φ *is latitude,* λ *is longitude, R is earth's radius (mean radius = 6,371km); note that angles need to be in radians to pass to trig functions!*

Fig 6 Haversine formula

Haversine formula pseudocode:

```
dlon = lon2 - lon1

dlat = lat2 - lat1

a = (sin(dlat/2))^2 + cos(lat1) * cos(lat2) * (sin(dlon/2))^2

c = 2 * atan2( sqrt(a), sqrt(1-a) )

d = R * c (where R is the radius of the Earth)

R = 6367 km OR 3956 mi
```

Fig 7 Haversine formula pseudocode

This optimized list of clients for each caregiver after applying the many-to-one Gale Shapley algorithm will be the input to the scheduling algorithm.

4.2 State 2: Scheduling Engine

The scheduling model has the goal of developing the weekly plan (i.e., day and time/order) of visits for each caregiver, taking as input the output of the assignment model.

4.2.1 Distance Heuristic

The distance heuristic [15] is a greedy method for assigning a new request between two patients when the insertion cost/additional travel time is the minimum. The cost is computed by subtracting the distance between a request's predecessor and successor from the sum of the request's predecessor and successor distances. The insertion cost, C, is computed as follows: If the Euclidean distance between a request and its predecessor and successor is k_1 and k_2, and the Euclidean distance between the predecessor and its successor is k_3, the insertion cost, C, can be calculated as follows:

$$C = k_1 + k_2 - k_3$$

As a result, whenever a new patient enters the system, the algorithm evaluates the cost of insertion of that patient among all requests previously allocated on each day of the week, if intervals are possible. The technique then chooses the cheapest interval in a day/days based on the patient's visit frequency. Finally, within the patient's service horizon, all visits are arranged on the cheapest days and times. The appointment time is determined by the request's closeness to the predecessor/successor.

For example, if the distance between the request and the predecessor is shorter than the distance between the request and the successor, the visit will begin immediately following the predecessor's visit, allowing for sufficient travel time. If two days have the same insertion expenses, we allocate the visit to the day with the fewest patient visits to balance the burden.

4.2.2 Capacity heuristic

Even if traveling from one appointment to the next involves more than one-time unit, the distance heuristic schedules them adjacent to one other. In such circumstances, allowing for a wider time gap between appointments may be useful, allowing for subsequent patients to be inserted in between without needing additional travel time.

When the trip duration is longer than a time slot, the capacity heuristic [15] prevents scheduling a new patient right next to an existing patient. Suppose any new patient is more than one-time slot distant from other patients on the schedule. In that case, the capacity heuristic allocates it to a time slot that allows enough time between it and its predecessor and successor patients to assign a future request.

For example, let's assume that the transit time between a new request and its predecessor (8.00 am) and successor (11.00 am) is 19 and 24 minutes, respectively, and that the service time for each is 30 minutes. 9.0, 9.15, 9.3, 9.45 & 10.00 are, therefore, candidate time windows. He is scheduled at 9.00 a.m. if we utilize the distance heuristic. In this instance, we may only book one more request at 9.45, 10.00, or 10.15 a.m. If we plan the request at 9.30, we have the option of scheduling two additional patients at 8.45 &10.15 if they only require a one-time slot to travel between their predecessors and successors. As a result of the capacity heuristic, future patients will face gaps. Of course, there must be ample time between predecessor and successor patients to accommodate the present need. Otherwise, requests are assigned according to the distance heuristic.

4.2.3 Travelling Salesman Problem

One of the most popular problems in combinatorial optimization is the Traveling Salesman Problem (TSP). In the disciplines of computer science and operations research, this is a well-known algorithmic issue. TSP refers to the task of determining the quickest and most efficient route for a person given a set of particular destinations. There are clearly many options, but mathematicians and computer scientists have spent decades attempting to solve the problem of identifying the optimal one—the one that requires the least distance or expense.

If there are n cities, there are (n - 1)! Possible ways for his tour. For example, if the number of cities to be visited is 5, then there are 4! different combinations.

The objective of TSP is to find a solution if a travelling salesman wants to visit his customers located in different cities. He starts in one of the cities and returns thereafter, having paid all the visits. The TSP algorithm aids in determining which cities the salesman should visit in order to minimize overall journey distance.

Popular Traveling Salesman Problem Solutions

i) The Brute-Force Approach

To find the shortest unique solution, the Brute Force technique, also known as the Naive Approach, calculates and analyses all conceivable permutations of routes or pathways. To use the Brute-Force method to solve the TSP, you must first determine the total number of routes, then draw and list all of them. Calculate the distance between each route and then select the one that is the shortest—this is the best option.

ii) The Branch and Bound Method

This strategy divides an issue into sub-problems to be solved. It's a method for resolving a set of sub-problems, each of which may have numerous viable solutions, and where the solution chosen for one sub-problem may influence the answers to future sub-problems. To use the Branch and Bound approach to solve the TSP, you must first pick a start node and then set it bound to an extremely big number (let's say infinity). Choose the shortest path between the unvisited node and the current node, then multiply the distance by the current distance. If the current distance is less than the bound, repeat the operation. You're done if the current distance is less than the bound.

This strategy divides an issue into smaller chunks that may be tackled individually. It's a method for resolving a set of sub-problems, each of which may have numerous alternative solutions, and where the solution chosen for one sub-problem may influence the answers to future sub-problems. To use the Branch and Bound approach to solve the TSP, pick a start node and then set it bound to an extremely big value (let's say infinity). Choose the shortest arch between the unvisited and current nodes, then multiply by the current distance. Continue until the current distance is less than the bound. You're done if your current distance falls within the limit. You can combine the distances together to make the bound match the current distance. Continue in this manner until all of the arcs are covered.

iii) The Nearest Neighbor Method

This is the most basic TSP heuristic. The key to this strategy is to always visit the closest destination before returning to the first city after seeing all of the others. Choose a random city to solve the TSP with this approach, then seek for the closest unvisited city and travel there. You must return to the first city once you have visited all of the cities.

4.2.4 Scheduling Algorithm

We are using a modified version of the TSP for scheduling visits as every client needs to be visited by the caregiver at least once a week with his daily route starting from and ending at his house.

Objective Function

Minimize the amount of time a caregiver has to work each day.

- Using the output from the assignment model, a travel time matrix is created for each caregiver with only the clients that he/she needs to service.

- If a caregiver '**x**' from discipline '**Y**' has '**a**' clients to service, a travel time matrix

  t[Y][x] is defined as a (a+2)*(a+2) matrix,

  where t[Y][x][i][j] = Time required for caregiver to travel from Client '**i**' to Client '**j**'.

- Where i=0 and i = a+1 corresponds to the caregiver's house itself while i = 1,2,...,a corresponds to the ‘a’ clients that the caregiver needs to service.

- Thus, every caregiver has to start from i=0 service the clients and return to i=a+1.

Constraints

- Every client can be visited by a caregiver at most once every day.
- Each Caregiver should go to exactly 1 Client’s house from his/her house.
- Each Caregiver should return to his/hers house from exactly 1 Client’s house.
- Flow Conservation: If a Caregiver visits a client, he must also exit it.
- No Subtours should be formed, i.e., a caregiver should have only one route every day and not more than one route.
- Non-Linear Constraint that calculates arrival time to each client's house by Caregiver(This is converted into a linear constraint).

Decision Variables

- $$VAR_{ijd} = \begin{cases} 1; & \text{if Caregiver goes from ‘i’ to ‘j’ on day ‘d’} \\ 0; & \text{otherwise} \end{cases}$$
- U_{id} : This variable stops subtours.
- Z_{id} : Amount of time client ‘i’ is serviced on day ‘d’.
- Y_{d} : Amount of time caregiver has to work on day ‘d’.
- $DUMMY_{id}$: Dummy variable used to calculate Y_{d}.

4.3 Gurobi Optimizer and PuLP

4.3.1 Gurobi Optimizer

The fastest and one of the most powerful mathematical programming solvers available for Linear Programming (LP), Quadratic programming (QP), and MIP (MILP, MIQP, and MIQCP) problems. Gurobi outperforms competitors in terms of finding both viable and proven optimum solutions (including CPLEX and XPress). As model size and difficulty expand, the performance difference widens.

Gurobi's services are available in a variety of languages, including JAVA, Python, C++, and others. They also provide a speed that is up to 50% quicker than the nearest competition, as well as resilience that has been tested on a wide range of demanding issues from various sectors. Gurobi, on the other hand, is compensated and requires licensure.

4.3.2 PuLP Linear Programming

The Python library for linear optimization (PuLP) is an open-source linear programming (LP) software that employs Python syntax and includes a number of industry-standard solvers. It also integrates nicely with a range of open source and commercial LP solvers.

We use Gurobi Optimizer and PuLP to implement the scheduling model. Both Gurobi Optimizer and PuLP linear programming use the same assignment paradigm. However, we try to produce an optimal output using Gurobi Optimizer and produce a heuristic result using PuLP linear programming.

CHAPTER 5

RESULTS

Simulated data comprises **107** caregivers serving customers at various locations (latitude, longitude) and has **748** separate clients with **1134** authorization requests in total, with randomly generated time periods and disciplines from when they want the service until when, and a weekly restriction of merely the number of units the client wants to be serviced each week.

We simulated six distinct disciplines: SN, ST, PT, CNA, MSW, and OT, with randomized client demands for each field.

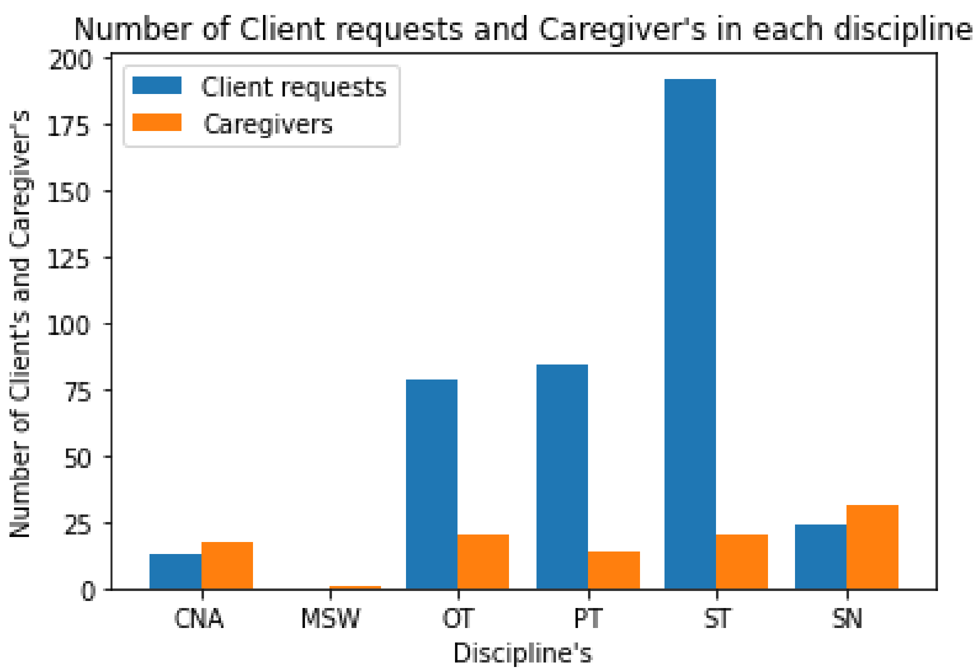

Fig 8 Number of client requests and caregivers in each discipline

The simulated caregiver's data includes unique caregiver IDs, locations in terms of latitude and longitude, and the skill/discipline service they perform. Clients primarily have client IDs, unique authorization IDs (because a single client may request multiple services), the date from and to which the client wishes to claim a service, and the unused and total number of credits (which indicates whether they are eligible to claim the service), and their location in terms of latitude and longitude.

We attempted to cover all scenarios, such as when client requests and caregivers are almost equal when client requests are far larger than caregivers accessible to service the clients, and certain edges where we have caregivers but no client demands for that discipline.

5.1 Gale Shapley Optimization

We utilize two sets of preferences and apply the many-to-one Gale Shapley algorithm to eliminate randomness and obtain an optimum prospective list of clients that a caregiver may service. Because one client may have one caregiver at the top of their preference list, that does not guarantee the caregiver would have that client at the top of theirs.

Looking at few output we achieved,

Potential list of clients for each caregiver without applying Gale Shapley algorithm:
{ 4: [22], 8: [22], 27: [375, 340, 453, 430, 264, 187],
37: [319, 341, 255, 47, 399, 342], 80: [424, 425, 355, 74, 452, 388], 98: [407, 523, 544, 318],103: [407, 523, 544, 318]... }

Potential list of clients for each caregiver after applying Gale Shapley algorithm:
{ 4: [56], 8: [55], 27: [375, 340, 453, 430, 264, 114],
37: [319, 255, 47, 342, 381, 8], 80: [424, 425, 355, 261, 103, 404], 98: [407, 523, 544, 318], 103: [34, 12, 497, 143]... }

We observe several differences before and after applying the Gale Shapley algorithm. To see how big of a difference Gale Shapley makes, we compare the similarity percentages of the preference lists created before and after Gale Shapley for each caregiver.

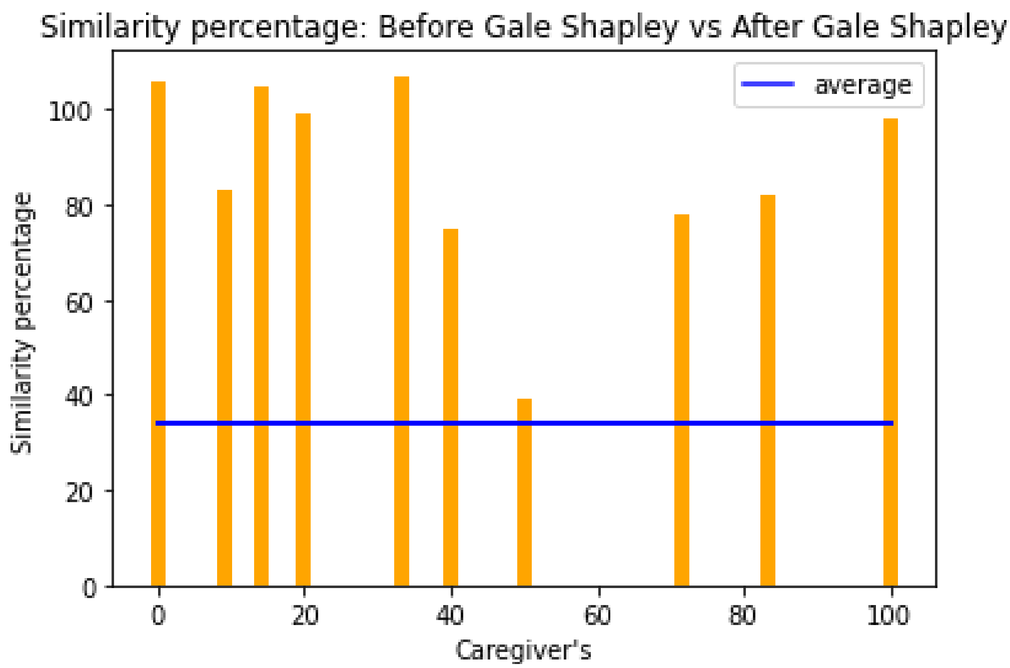

Fig 9 Similarity percentage: Before and After applying Gale Shapley

We calculate the similarity percentage by comparing the preference lists created before and after Gale Shapley for each caregiver i.e

$$f(Xi, Yi) = \begin{cases} 1; & \text{if Xi=Yi} \\ 0; & \text{otherwise} \end{cases}$$

$$Similarity\ \% = (\sum_{i=1}^{n} f(Xi, Yi) \times 100)/n$$

We can observe that there is a **34.17%** similarity percentage before and after Gale Shapley, implying that preferences alter by around **65.83%** after taking the client's and caregiver's preferences into account.

So using Gale Shapley results in more accurate mapping of possible client lists for each caregiver, eliminating randomness and allowing the caregiver to create an optimal prospective client list.

After the literature review, one of our focuses was on resource utilization, i.e., caregiver utilization. This is because the HC industry must adhere to labor norms when it comes to employment, and no caregivers should be overworked while other caregivers are available to assist the clients.

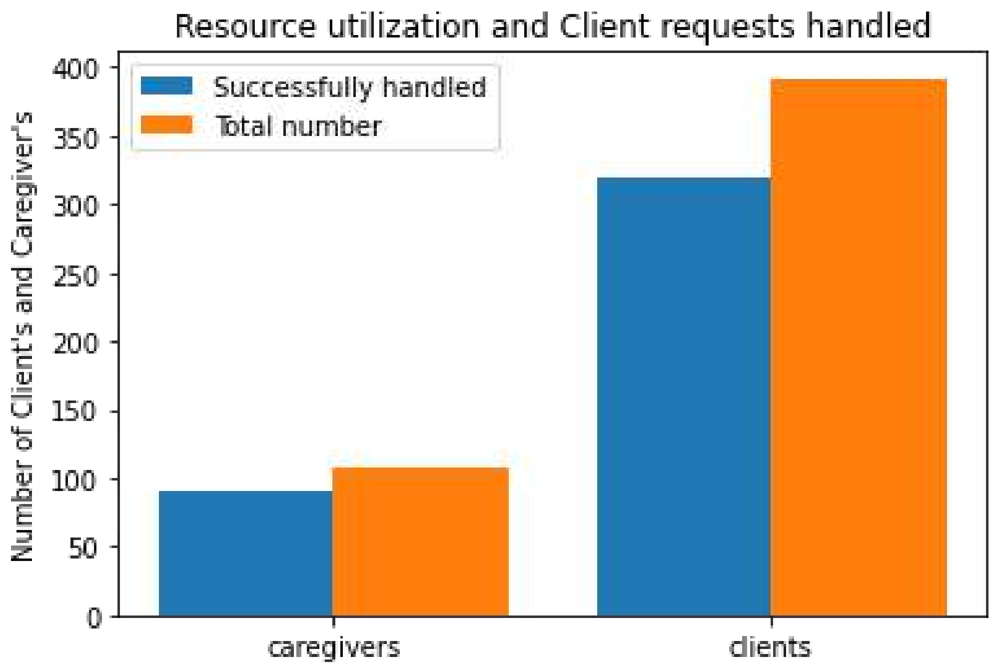

Fig 10 Resource utilization and client requests handled

In the assignment model, we were able to utilize approximately up to **82%** of the caregivers. Because of the edge case consideration, a few caregivers have been ruled out. And we were able to handle **78%** of client requests.

The overall processing time for the assignment model was under **2.2 seconds**.

5.2 Scheduling Optimization

We explored how greedy algorithms like distance heuristics and capacity heuristics operate in previous sections. With various constraints and decision variables, our scheduling engine's major goal was to reduce the amount of time a caregiver had to work each day, making the model optimum.

5.2.1 Major constraints

- Every client can be visited by a caregiver at most once every day

$$\sum_{j=0}^{a+1} VAR_{ijd} = 1$$

where, j: clients, a: number of clients, d: day, VAR_{ijd}: Caregiver goes from ‘i’ to ‘j’ on day ‘d’;

- Each Caregiver should go to exactly 1 Client’s house from his/hers house
- Each Caregiver should return to his/hers house from exactly 1 Client’s house

$$\sum_{j=1}^{a} VAR_{0jd} = 1; \quad \sum_{i=1}^{a} VAR_{i(a+1)d} = 1$$

where, j: clients, a: number of clients, d: day, VAR_{0jd}: caregivers go from ‘0’ his house to a particular client;

where, i: clients, a: number of clients, d: day, $VAR_{i(a+1)d}$: caregivers go from ‘particular i’ client to his house;

- Flow Conservation : If a Caregiver visits a client, he must also exit it

$$\sum_{i=0}^{a+1} VAR_{ijd} - \sum_{i=0}^{a+1} VAR_{jid} = 0$$

where, i: clients, j: particular client, a: number of clients, d: day, VAR_{ijd}: Caregiver goes from ‘i’ to ‘j’ on day ‘d’, VAR_{jid}: Caregiver goes from ‘j’ to ‘i’ on day ‘d’;

- No Subtours should be formed, i.e., a caregiver should have only one route every day and not more than one route

$$u_{id} - u_{jd} + a * (VAR_{ijd}) \leq (a - 1)$$

where, d: days, i: clients, j: clients, u_{id}: This variable stops subtours, u_{jd}: This variable stops subtours, a: number of clients, VAR_{ijd}: Caregiver goes from ‘i’ to ‘j’ on day ‘d’;

By subtours, we mean that caregivers are not allowed to take pauses between schedules. For example, if a caregiver is scheduled to service [1, 2, 3, 4], he cannot service 1, 2, then take a break before returning to serve 3, 4.

- Non-Linear Constraint that calculates arrival time to each client's house by Caregiver (This is converted into a linear constraint).

$$y_d = \sum_{i=0}^{a+1} \sum_{j=1}^{a+1} ((t_{Yxij} + z_{jd}) * (VAR_{ijd}))$$

Where, y_d: the number of hours of work on a particular day ‘d’, j: client, i: client, t_{Yxij}: ‘t’ travel time matrix, Y: Discipline, x: caregiver ID, z_{jd}: number of hours of service a client receives, VAR_{ijd}: Caregiver goes from ‘i’ to ‘j’ on day ‘d’;

5.2.2 Gurobi Optimizer vs. PuLP

We have implemented the scheduling model using Gurobi Optimizer and PuLP, below are graphs comparing results of the same.

To comply with the labor regulations of the HHC industry, we have established an upper limit of 8 hours per day for caregivers to work, and we are charting overtimes below for each model, which are the number of caregivers that need to work more than 8 hours per day to complete client requests. In Gurobi Optimal code, we establish an objective function to reduce the amount of time a caregiver needs to work each day, but we don't set one for the Gurobi Heuristic and PuLP Heuristic. The backend of the Gurobi and PuLP solves the equation to fulfil our requirements in the Heuristic models, however the result is not optimal.

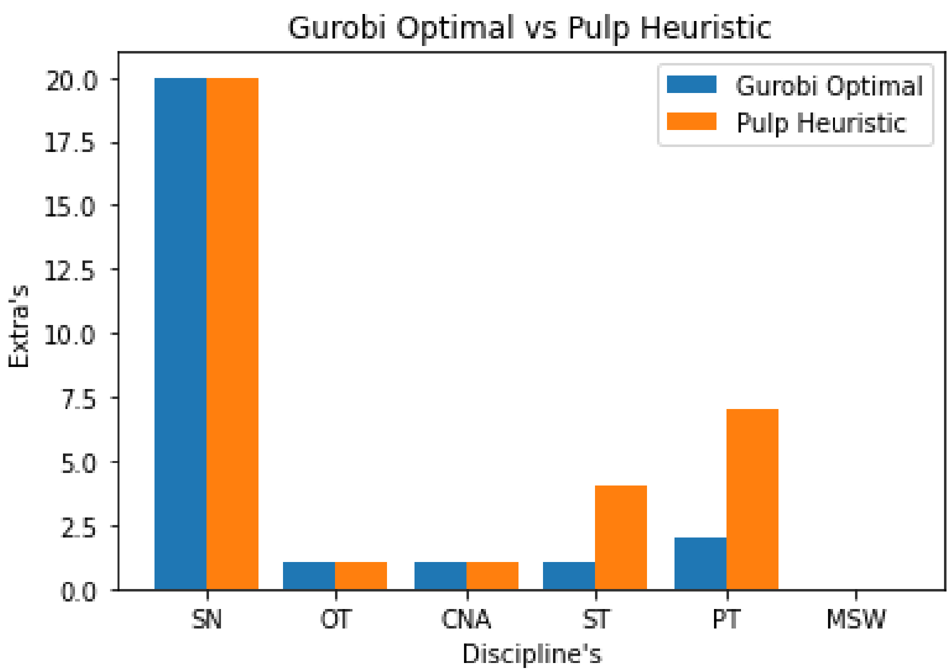

Fig 11 Gurobi Optimal vs. Pulp Heuristic

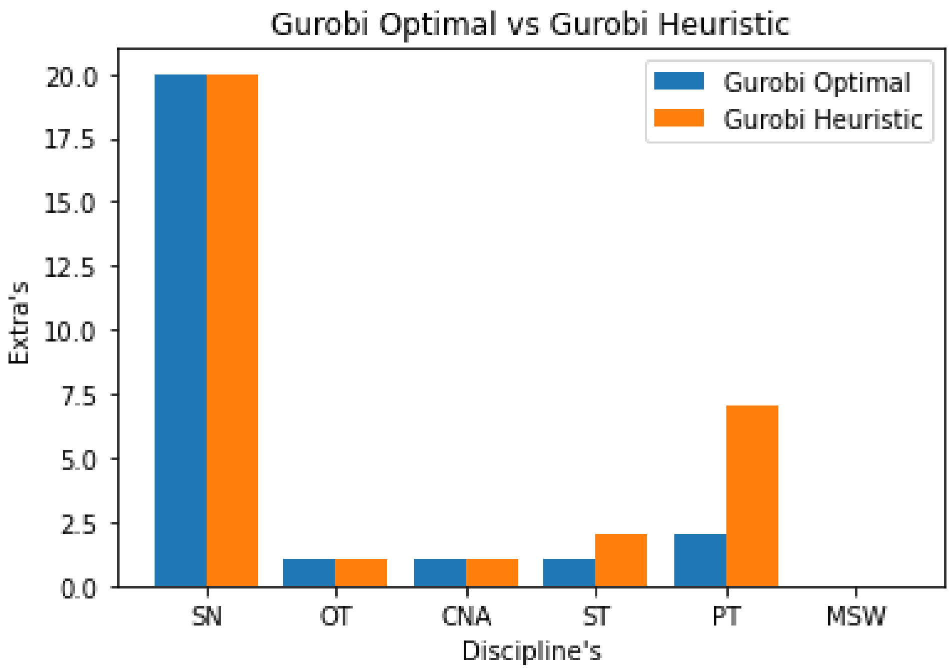

Fig 12 Gurobi Optimal vs. Gurobi Heuristic

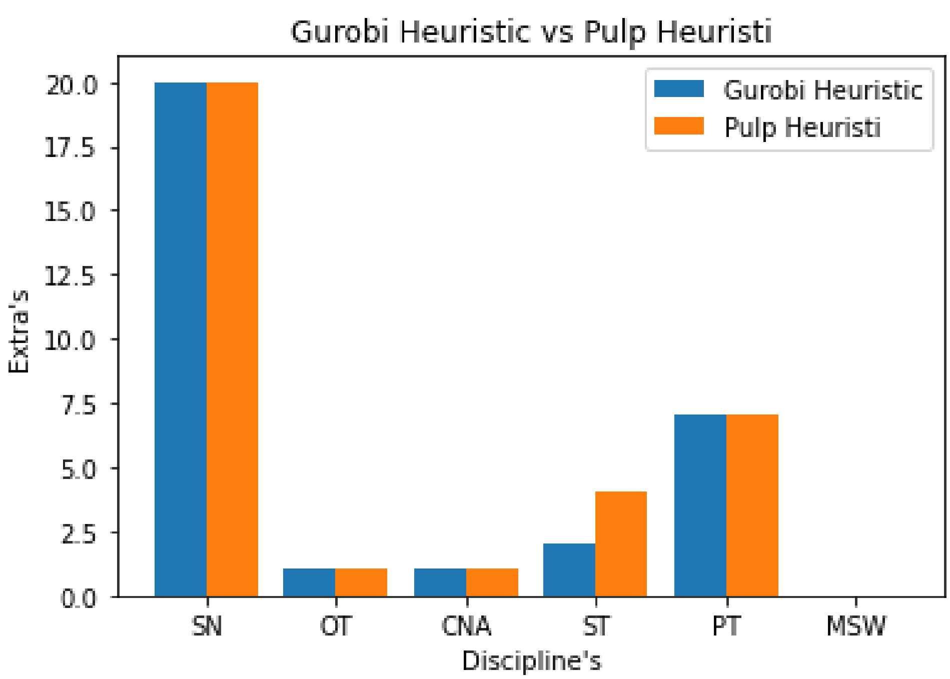

Fig 13 Gurobi Heuristic vs. PuLP Heuristic

Table 7. Model wise overtimes

Model	SN	OT	CNA	ST	PT	MSW
Gurobi Optimal	20	1	1	1	2	0
Gurobi Heuristic	20	1	1	2	7	0
Pulp Heuristic	20	1	1	4	7	0

Because these are overtimes, i.e these caregivers are required to work more than 8 hours per day, HHC agencies must manually select which clients they want the caregiver to serve, or if there is a predetermined preference, we may take it into account, which we haven't done yet.

Alternatively, we may apply the greedy method to these extras and simply map them to top clients using the greedy approach.

Else caregivers who work overtime should be compensated, and their hours should be mapped to clients following consultation with the caregivers.

- All clients and caregivers given from the Assignment model are being utilized in the scheduling model
- Gurobi Optimal Model outperforms the other two models, Gurobi Heuristic and Pulp Heuristic, in terms of scheduling
- The schedules are computed in roughly 7 seconds by Gurobi Optimal, 3 seconds by Gurobi Heuristic, and 2 minutes 14 seconds by Pulp Heuristic
- Pulp Heuristic takes the highest time to compute

CHAPTER 6

CONCLUSION

Importance of home healthcare (HHC) is growing day by day since populations of developed and even developing countries are getting older quickly, and the number of hospitals, retirement homes, and medical staff does not increase at the same rate.

An HC provider's resource short-term planning process necessitates adherence to a wide number of limitations and objectives, both in terms of system efficiency and care quality. This is especially true for short-term human resource planning. So, as a result, we focused our research on creating an automated support tool for healthcare organizations to help them with the same. The solution is hierarchical and consists of two linear programming models: the first deals with assigning clients to a reference caregiver, and the second is a scheduling model whose output is the weekly plan for each caregiver.

This paradigm can provide several benefits during the scheduling process. First and foremost, the service quality and, as a result, patient happiness may be increased. Second, rather than utilizing greedy or first come, first serve algorithms, we do two levels of optimization, one using the assignment model and the other using the scheduling model, which gives us superior scheduling outcomes. Third, and most crucially, the time and the manpower-intensive job of scheduling caregivers to patients is optimally automated, decreasing scheduling time and the need for manpower. So that healthcare providers can maximize their patient care activities while reducing the required effort and costs by ensuring optimized allocation of patient care activities to their scarce resources.

Future research could also include the use of different techniques for demand forecasting and for scenario generation to assess the impact of demand estimation and industrial accuracy of the solutions obtained with the two-stage programming problem.

Another thing to consider is revenue management. In such a situation, patient preferences for visit days and times are assessed, and appropriate charges are calculated based on the best schedules. Furthermore, due to the intricacies of some patient visits, more than one nurse is required, and optimizing nurse routing and scheduling under this limitation appears to be both fascinating and challenging for future studies.

APPENDIX A

The final results after applying assignment and scheduling models on the simulated data we attain a weekly schedule for each caregiver in that order they are going to service the list of clients over the week.

The following results have all the available caregivers discipline wise.
The output looks like the following:

```
{
‘CNA’ :
            { CG1 :
                        { 0:[Cl1,Cl2,..],
                          1:[CL,Cl2,...], ...
                          6:[Cl1,Cl2,...] },
              CG2 : {...},...
             },
‘ST’ : {...}, …
}
```

This is to be interpreted as follows: CG1 in CNA discipline should visit (Cl1,CL2..) in this particular order on Day0.

Scheduling Model Output example:

{'SN': {
3: {0: [527], 1: [527], 2: [527], 3: [527], 4: [527], 5: [527], 6: [527]},
4: {0: [56], 1: [56], 2: [56], 3: [56], 4: [56], 5: [56], 6: [56]},
8: {0: [55], 1: [55], 2: [55], 3: [55], 4: [55], 5: [55], 6: [55]},
9: {0: [2], 1: [2], 2: [2], 3: [2], 4: [2], 5: [2], 6: [2]},
10: {0: [328], 1: [328], 2: [328], 3: [328], 4: [328], 5: [328], 6: [328]}},
'OT': {67: {0: [65, 462, 480, 299],
1: [480, 299, 462, 65],
2: [480, 299, 462, 65],
3: [480, 299, 462, 65],
4: [299, 480, 462, 65],
5: [480, 299, 462, 65],
6: [65, 462, 480, 299]},
6: {0: [151, 457, 329, 434],
1: [434, 329, 457, 151],
2: [434, 329, 457, 151],
3: [151, 457, 329, 434],
4: [434, 329, 457, 151],

5: [434, 329, 457, 151],
6: [434, 329, 457, 151]},
71: {0: [175, 277, 32, 37],
1: [37, 32, 277, 175],
2: [37, 32, 277, 175],
3: [37, 32, 277, 175],
4: [37, 32, 277, 175],
5: [37, 32, 277, 175],
6: [175, 277, 32, 37]},
73: {0: [161, 8, 47, 221],
1: [161, 8, 47, 221],
2: [8, 161, 47, 221],
3: [8, 161, 47, 221],
4: [8, 161, 47, 221],
5: [221, 47, 161, 8],
6: [221, 47, 161, 8]},
75: {0: [216, 216, 269, 238],
1: [216, 216, 269, 238],
2: [216, 216, 269, 238],
3: [238, 269, 216, 216],
4: [216, 216, 269, 238],
5: [238, 269, 216, 216],
6: [238, 269, 216, 216]}},
'CNA': {
13: {0: [433], 1: [433], 2: [433], 3: [433], 4: [433], 5: [433], 6: [433]},
25: {0: [249], 1: [249], 2: [249], 3: [249], 4: [249], 5: [249], 6: [249]},
28: {0: [3], 1: [3], 2: [3], 3: [3], 4: [3], 5: [3], 6: [3]},
29: {0: [156], 1: [156], 2: [156], 3: [156], 4: [156], 5: [156], 6: [156]},
32: {0: [223], 1: [223], 2: [223], 3: [223], 4: [223], 5: [223], 6: [223]}},
'ST': {64: {0: [32, 298, 308, 436, 420, 37],
1: [32, 37, 436, 298, 420, 308],
2: [37, 436, 420, 32, 308, 298],
3: [308, 436, 420, 32, 37, 298],
4: [32, 308, 37, 420, 436, 298],
5: [32, 298, 436, 308, 37, 420],
6: [436, 32, 420, 308, 37, 298]},
69: {0: [226, 267, 151, 74, 457, 53],
1: [226, 53, 74, 267, 457, 151],
2: [53, 74, 457, 226, 151, 267],
3: [151, 74, 457, 226, 53, 267],
4: [226, 151, 53, 457, 74, 267],
5: [226, 267, 74, 151, 53, 457],
6: [74, 226, 457, 151, 53, 267]},
70: {0: [85, 93, 238, 135, 269, 125],
1: [85, 125, 135, 93, 269, 238],
2: [125, 135, 269, 85, 238, 93],
3: [238, 135, 269, 85, 125, 93],
4: [85, 238, 125, 269, 135, 93],
5: [85, 93, 135, 238, 125, 269],
6: [135, 85, 269, 238, 125, 93]},
72: {0: [335, 229, 113, 426, 213, 284],

1: [335, 284, 426, 229, 213, 113],
2: [284, 426, 213, 335, 113, 229],
3: [113, 426, 213, 335, 284, 229],
4: [335, 113, 284, 213, 426, 229],
5: [335, 229, 426, 113, 284, 213],
6: [426, 335, 213, 113, 284, 229]},
12: {0: [224, 348, 460, 170, 10, 29],
1: [224, 29, 170, 348, 10, 460],
2: [29, 170, 10, 224, 460, 348],
3: [460, 170, 10, 224, 29, 348],
4: [224, 460, 29, 10, 170, 348],
5: [224, 348, 170, 460, 29, 10],
6: [170, 224, 10, 460, 29, 348]}},
'PT': {65: {0: [45, 478, 154, 173, 363, 310],
1: [45, 310, 173, 478, 363, 154],
2: [310, 173, 363, 45, 154, 478],
3: [154, 173, 363, 45, 310, 478],
4: [45, 154, 310, 363, 173, 478],
5: [45, 478, 173, 154, 310, 363],
6: [173, 45, 363, 154, 310, 478]},
66: {0: [273, 368, 258, 247, 220, 455],
1: [273, 455, 247, 368, 220, 258],
2: [455, 247, 220, 273, 258, 368],
3: [258, 247, 220, 273, 455, 368],
4: [273, 258, 455, 220, 247, 368],
5: [273, 368, 247, 258, 455, 220],
6: [247, 273, 220, 258, 455, 368]},
68: {0: [151, 56, 295, 75, 53, 459],
1: [151, 459, 75, 56, 53, 295],
2: [459, 75, 53, 151, 295, 56],
3: [295, 75, 53, 151, 459, 56],
4: [151, 295, 459, 53, 75, 56],
5: [151, 56, 75, 295, 459, 53],
6: [75, 151, 53, 295, 459, 56]},
74: {0: [467, 149, 460, 346, 489, 12],
1: [467, 12, 346, 149, 489, 460],
2: [12, 346, 489, 467, 460, 149],
3: [460, 346, 489, 467, 12, 149],
4: [467, 460, 12, 489, 346, 149],
5: [467, 149, 346, 460, 12, 489],
6: [346, 467, 489, 460, 12, 149]},
107: {0: [485, 124, 204, 413, 414, 469],
1: [485, 469, 413, 124, 414, 204],
2: [469, 413, 414, 485, 204, 124],
3: [204, 413, 414, 485, 469, 124],
4: [485, 204, 469, 414, 413, 124],
5: [485, 124, 413, 204, 469, 414],
6: [413, 485, 414, 204, 469, 124]}},
'MSW': {}}

BIBLIOGRAPHY

[1] Health Transition Fund, Health Canada(2001), Costs of acute care and home care services, Report.

[2] World Health Organization, US National Institute of Aging. Global health and aging. Bethesda, Maryland: National Institutes of Health; 2011 (NIH Publication no. 11-7737).

[3] A strategy for active, healthy ageing and old age care in the Eastern Mediterranean Region 2006– 2015. Cairo: WHO Regional Office for the Eastern Mediterranean; 2006 (WHO-EM/HSG/030/E).

[4] Burke EK , De Causmaecker P , Berghe GV , Van Landeghem H(2004) . The state of the art of nurse rostering;7(6):441–99.

[5] Demirbilek M, Branke J, Strauss A (2018) Dynamically accepting and scheduling patients for home healthcare. Health Care Manag Sci. https://doi.org/10.1007/s10729-017-9428-0

[6] Valeria Borsani, Andrea Matta, Giacomo Beschi, Francesco Sommaruga(2006). A Home Care Scheduling Model For Human Resources;

[7] Redjem R, Marcon E (2016) Operations management in the home care services: a heuristic for the caregivers' routing problem. Flex Serv Manuf J 28(1–2):280

[8] Issabakhsh M, Hosseini-Motlagh SM, Pishvaee MS, Saghafi Nia M (2018) A vehicle routing problem for modeling home healthcare: a case study. Int J Transp Eng 5(3):211

[9]Guericke D, Suhl L (2017) The home health care problem with working regulations. OR Spectrum 39(4):977–1010

[10] Frifita S, Masmoudi M, Euchi J (2017) General variable neighborhood search for home healthcare routing and scheduling problem with time windows and synchronized visits. Electron Notes Discrete Math 58:63

[11] Wirnitzer J, Heckmann I, Meyer A, Nickel S (2016) Patient-based nurse rostering in home care. Oper Res Health Care 8:91

[12]Bertels S., Fahle T.(October 2006), A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem.

[13] Eveborn P., Flisberg P., Rönnqvist M.(2006), LAPS CARE—an operational system for staff planning of home care, European Journal of Operational Research , Vol 171
[14] Begur S.V., Miller D.M., Weaver J.R.(1997), An integrated spatial DSS for Scheduling and Routing Home-Health Care Nurses, pp.35-48.
[15] Bennett AR, Erera AL (2011) Dynamic periodic fixed appointment scheduling for home health. IIE Trans Healthcare Syst Eng 1(1):6–19
[16] Gale, D., L. S. Shapley (1962).College admissions and the stability of marriage, American Mathematical Monthly 69 9-15.