Dynamic Routing and Scheduling for Home Healthcare Services with Cooperating Multiple Service Providers

by

Mert Parçaoğlu

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

KOÇ ÜNİVERSİTESİ

August 8, 2022

Dynamic Routing and Scheduling for Home Healthcare Services with Cooperating Multiple Service Providers

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Mert Parçaoğlu

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. F. Sibel Salman (Advisor)

Prof. Özgür M. Araz

Prof. Lerzan Örmeci

Date:

To my family

ABSTRACT

Dynamic Routing and Scheduling for Home Healthcare Services with Cooperating Multiple Service Providers

Mert Parçaoğlu

Master of Science in Industrial Engineering

August 8, 2022

The importance of home healthcare services has been increasing globally with aging population and the rise of remote activities in our lives. In this thesis, we focus on a daily routing and scheduling problem for home healthcare services considering patient priorities defined according to their health conditions and the dynamic emergence of urgent patient visit requests during the day which is handled by the cooperation of multiple service providers. We optimize multiple objectives in a lexicographic goal programming (LGP) framework.

The LGP approach is used for both the static and dynamic problems. The solution to the static problem identifies an initial schedule and a routing plan for each healthcare service provider (HSP) separately. On the other hand, the dynamic problem optimizes the schedules and routes of all HSPs and schedules dynamic emergent patients in addition to the patients scheduled in the initial plan. By means of the dynamic problem, the routing and scheduling plans of the HSPs are re-optimized simultaneously each time an emergent patient arrives. For both problems, a mixed integer linear programming (MILP) model and a greedy randomized adaptive search procedure (GRASP), followed by a variable neighborhood search (VNS) procedure for local improvement, are developed together with a hybrid approach that utilizes both the MILP model and the proposed heuristic. The three approaches are tested through a case study that considers COVID-19 patients in a district of Istanbul. In order to show the effectiveness of cooperation and compare the proposed solution methods, several performance metrics are analyzed through extensive simulation runs. We observe that i) our proposed heuristic approach provides solutions with quality close to or better than those obtained by the MILP run within a time limit, while running in short times, and ii) the cooperation strategy improves system performance significantly in a dynamic setting.

ÖZETÇE

Yardımlaşan Çoklu Servis Sağlayıcılarıyla Evde Sağlık Hizmetleri için Dinamik Rotalama ve Çizelgeleme
Mert Parçaoğlu
Endüstri Mühendisliği, Yüksek Lisans
August 8, 2022

Evde sağlık hizmetlerinin önemi, yaşlanan nüfus ve hayatımızdaki uzaktan gerçekleştirilen faaliyetlerin artmasıyla birlikte küresel olarak artmaktadır. Bu tezde, birden fazla hizmet sağlayıcının yardımlaşma ile ele alınan, sağlık koşullarına göre belirlenen hasta öncelikleri ve gün içinde acil hasta ziyaret taleplerinin dinamik olarak ortaya çıkması dikkate alınarak, evde sağlık hizmetleri için günlük bir rotalama ve çizelgeleme problemine odaklanılmıştır. öncelikli hedef programlama (LGP) çerçevesinde birden çok hedefi en iyiliyoruz.

LGP yaklaşımı hem statik problem hem de dinamik problem için kullanılır. Statik problemin çözümü, her bir sağlık hizmeti sağlayıcısı (HSP) için ayrı ayrı olacak şekilde bir başlangıç çizelgesi ve bir rotalama planı tanımlar. Öte yandan, dinamik problem, tüm HSP'lerin çizelgelerini ve rotalarını en iyiler ve başlangıç planında çizelgelenen hastalara ek olarak dinamik acil hastaları çizelgeler. Dinamik problem ile birlikte, HSP'lerin rotalama ve çizelgeleme planları, her bir acil hasta geldiğinde hastanın gelişiyle eş zamanlı olarak yeniden en iyilenir. Her iki problem için, bir Karma Tamsayılı Doğrusal Programlama (MILP) modeli ve Açgözlü Rastgele Adaptif Arama Prosedürü (GRASP), ardından yerel iyileştirme için bir Değişken Komşuluk Arama (VNS) prosedürü, hem MILP modeli hem de önerilen sezgisel algoritmayı kullanan hibrit bir yaklaşımla birlikte geliştirilmiştir. Üç yaklaşım, İstanbul'un bir semtindeki COVID-19 hastalarını dikkate alan bir vaka çalışması ile test edilmiştir. Yardımlaşmanın etkinliğini göstermek ve önerilen çözüm yöntemlerini karşılaştırmak için kapsamlı simülasyon çalışmaları yoluyla çeşitli performans ölçütleri analiz edilir. i) önerilen sezgisel yaklaşımımızın, kısa sürelerde çalışırken, bir zaman sınırı içinde MILP çalışmasıyla elde edilenlere yakın veya daha iyi kalitede çözümler sağladığını ve ii) yardımlaşma stratejisinin dinamik bir ortamda sistem performansını önemli ölçüde iyileştirdiğini gözlemledik.

ACKNOWLEDGMENTS

I would like to express my endless thanks to my supervisor Prof. Sibel Salman who has always supported me throughout my master's study. She always inspired me with her work and her perspective on the subjects. I would like to thank Prof. Özgür M. Araz who I had the opportunity to work together on projects. I also would like to thank Prof. Lerzan Örmeci for her valuable feedbacks and time on my thesis.

I would like to endless thank Assist. Prof. Eda Yücel and Assoc. Prof. Nilgün Fescioğlu Ünver. I know that I couldn't have started my master's program journey without their support.

I would like to thank Yunus and Niyazi, with whom we have been continuing our adventures since high school, pushing the limits of our imagination. Also, I would like to thank to my undergraduate friends Berkay, Ece, Fatih Can, Diktaş, Boyacı, Sena, Duygu, and Ömer.

I would like to acknowledge TÜBİTAK for financially supporting me in the 2210-A scholarship program during my master's degree.

I would like to thank my colleagues from Borda Technology, especially my teammate Cansu and our team leader Burak for supporting me during the last year of the master's degree.

Finally, I would like to express my biggest thanks to my dear family, whose support I always felt in my heart. I want to send my love to my mother Saadet and my father Atila, who have been working since my birth to ensure a good life for me in the future. I can't say enough to thank my dear sister Yağmur and my brother-in-law Eray, who rushed to my calls for help during my most difficult times. Of course, I would also like to thank my uncle Salih, who is the main source of my interest in engineering.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

2-PHA	Two Phase Method
ALNS	Adaptive Large Neighborhood Search
aVNS	Adaptive Variable Neighborhood Search
B&P	Branch and Price Algorithm
CG	Column-Generation Algorithm
COVID-19	Coronavirus Disease 2019
DEC	Decomposition Method
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
HEU	Heuristic
HEU GRE	Greedy Heuristic
HHC	Home Healthcare
HHCRSP	Home Healthcare Routing and Scheduling Problem
HSP	Healthcare Service Provider
LGP	Lexicographic Goal Programming
MAT HEU	Matheuristic
MDLS	Multi-Directional Local Search
MILP	Mixed Integer Linear Programming
RA	Re-Optimization Approach
RCL	Restricted Candidate List
SA	Simulated Annealing
TS	Tabu Search
VNS	Variable Neighborhood Search

Chapter 1

INTRODUCTION

The increasing life expectancy of human beings has brought along an aging population and increased health problems as they get older. While the increase in health problems strains the healthcare systems, home healthcare systems emerge as a viable alternative, both because constantly going to the hospital is inefficient and unsustainable, and poses a threat in terms of epidemics and infectious diseases. When health expenditures in developed countries are taken into account, it is seen that home healthcare systems have an important share of 3%-5%, and they are increasing year by year [Nadine Genet and Saltman, 2012]. [Falvey et al., 2020] emphasize that any home-based interventions that would reduce the volume of hospitalizations, can contribute meaningfully to public health goals. Thus, it can be said that home healthcare plays a crucial role in public health. In fact, home healthcare systems have been used effectively in many countries, including Turkey, to control the Coronavirus Disease 2019 (COVID-19), which affected the whole world starting from 2020 [Spinelli and Pellino, 2020].

In this thesis, we focus on the operational decisions related to home healthcare services. We aim to optimize the healthcare service providers' (HSPs') daily routes and prioritized patient visit schedules dynamically as new emergency requests that should be serviced within the day arise. To handle this situation, particularly, we suggest the HSPs to cooperate with each other through central decision making. Each HSP has a list of patients that reside relatively closely and they normally visit only these patients for continuity of care purposes. However, when additional requests arrive during the day, in order to serve all the requests timely and to keep the overtime and the number of patients with postponed visits limited, an HSP may attend to the patient of another HSP so that their joint time is utilized better. In addition, considering an urban region, we incorporate time-dependent travel distances with respect to the changing traffic patterns within a day. Furthermore, we consider multiple objectives in lexicographic order in a goal programming framework.

In our setting, each healthcare service provider (HSP) is assigned a patient cluster, and for each of them an initial daily plan to visit patients selected from their assigned cluster at the beginning of the day is created. In this static problem, each HSP is in charge of his patient list for continuity of care purposes. On the other hand, when the more realistic dynamic case is considered, while the initial plan is being executed for each HSP, dynamically occurring urgent patient requests arrive. The studies in the literature follow an approach where these requests are assigned to the HSP of the related cluster and are included in the plan of the HSP. However, these changes in the initial plan may cause HSPs to postpone patient visits scheduled in the initial plan, change the visit times, and perform overtime. In order to avoid these problems, we propose a joint optimization problem for all HSPs, where HSPs may cooperate and serve patients from another HSP's list. For this purpose, we formulate a model that re-optimizes all routes according to multiple objectives in a lexicographic goal programming framework. The objectives are to maximize the total priority of the visited patients, minimize the total arrival time changes in the schedule, minimize the total overtime of the service providers, and minimize the total travel time of the service providers. In addition, we consider time-dependent travel times.

First, a mixed integer linear programming (MILP) model is developed for the static problem that generates an initial plan. However, since the model can handle only instances with a small number of patients, we develop a greedy randomized adaptive search procedure (GRASP), followed by a variable neighborhood search (VNS) procedure for local improvement. Then, for the dynamic problem that is solved every time a new patient request arises, we develop an MILP, again a GRASP and VNS heuristic, and also propose a hybrid solution approach utilizing both the model and the heuristic. We test the performance of the solution approaches via computational experiments based on a case study that considers COVID-19 patients in a district of Istanbul. We analyze some key performance metrics over different parameter settings and instance sizes via simulation runs. We summarize the performances of the proposed approaches, namely, MILP and GRASP+VNS for the static problem, in addition to the performances of the three approaches, namely, MILP, GRASP+VNS and hybrid, for the dynamic problem. We also quantify the benefits from HSPs' cooperation by a comparison to the model solution that did not allow cooperation.

Our main contribution in this work is the centralized generation of multiple HSPs' schedules and routes in a dynamic setting that accounts for urgent visit requests with the cooperation of HSPs While ensuring care continuity and incorporating time-dependent travel times, which is addressed for the first time in the literature to the best of our knowledge. We show the benefits of such an approach as opposed to generating the schedule of each HSP separately, as done in most studies in the literature, by means of our computational study. We also present a simulation study conducted on instances with different sizes and varying parameters that lead to several important insights for decision-makers.

In the next chapter, we examine the related literature and identify the distinguishing aspects of our problem from the well-known problems and state our contributions. In Chapter 3, we explain the dynamic problem and the solution approaches we offer in detail. We describe the proposed re-optimization approach, as well as the two heuristic approaches, and the simulation framework in Chapter 4. Chapter 5 presents the data generation steps and results of the simulation study. Finally, Chapter 6 presents our concluding remarks.

Chapter 2

LITERATURE REVIEW

In this chapter, we will examine the problems studied in the home healthcare literature in terms of their setting, objectives, and solution methods. Firstly, the literature review table, which includes the articles in the HHCRSP field and the features examined in these articles, is presented. The table describes how the features are covered in the literature. In the next section, the contributions of this thesis to the literature are mentioned.

2.1 Literature Review Table

We can first classify the problems as static and dynamic. In the static problems, the patients to be visited are known in the planning phase, and while the plan is being followed, there are no changes in the conditions of the patients, employees, or environmental factors, beyond what is expected. Some studies include stochastic times in the static problem, generally reflecting the uncertainties in the service times and the travel times [Yuan et al., 2018, He and Shi, 2019, Shi et al., 2019, Liu et al., 2019b, Hosseinpour-Sarkarizi et al., 2020].

The focus in the literature has been on the static problem in general, hence studies on the dynamic problem is limited in number. However, there is an increasing interest in the dynamic problem as it can be seen from the features of the studies summarized in Table 2.1.

In the dynamic problems, the models obtain the solution by occurrence of the changes, such as coming new patients and cancellations, [Hertz and Lahrichi, 2009, Nickel et al., 2012, Fikar et al., 2016, López-Santana et al., 2016, Yuan and Jiang, 2017, Ouertani et al., 2019, Demirbilek et al., 2019, Hassen et al., 2019, Du et al., 2019, Demirbilek et al., 2021]. Also, there are models which work with predefined times [Bennett and Erera, 2011, Xie and Wang, 2017]. These models wait until the predefined times and work if a new solution is needed in the valid time interval. The definitions of dynamism differ among the articles, and it can be exemplified as the emergence of new patients [Bennett and Erera, 2011, Nickel et al., 2012, Ouertani et al., 2019, Demirbilek et al., 2019], the cancellation of patient requests [Fikar et al., 2016, Xie and Wang, 2017], the changes on the time windows of the patients [Yuan and Jiang, 2017] or dynamic changes in the number of healthcare service providers [Nasir and Dang, 2018].

As can be seen in Table 2.1, many different features have been incorporated to the problem variants studied. Time windows indicate that each patient has an assigned visit start/end time that must be obeyed if the time windows are defined as hard constraints. When soft time window constraints exist, the deviation from the predefined time window is penalized [Decerle et al., 2019, Decerle et al., 2018a, Mascolo et al., 2018, Decerle et al., 2017, Decerle et al., 2018b, Nickel et al., 2012]. In some cases, due to high demand, all of the patients might not be visited within the planning horizon [Nickel et al., 2012, Manerba and Mansini, 2016, Ly Nguyen and Montemanni, 2016, Liu et al., 2017, Yuan and Jiang, 2017, Yuan et al., 2018, Decerle et al., 2018b, Xiao et al., 2018, Nasir and Dang, 2018, He and Shi, 2019, Heching et al., 2019, Demirbilek et al., 2019, Hassen et al., 2019, Cinar et al., 2021, Liu et al., 2019b, Grenouilleau et al., 2019, Chaieb et al., 2020, Hosseinpour-Sarkarizi et al., 2020, Fathollahi-Fard et al., 2021, Demirbilek et al., 2021]. Several studies include the decision on whether the new patients will be accepted or will not [Demirbilek et al., 2019, Demirbilek et al., 2021, Bennett and Erera, 2011, Nickel et al., 2012, Nasir and Dang, 2018]. Additionally, the postponement of the patients on the visiting list has been considered in the literature [Liu et al., 2019a].

Table 2.1: Previous Survey Comparison

		Type		Features												Goals			
Author(s)	**Method**	**Static**	**Dynamic**	**Time Windows**	**Unvisited**	**Emergent**	**Multiperiod**	**Multi HSP**[1]	**Multi skills**	**Time-dependent TT**[2]	**Uncertain TT**[3]	**Uncertain ST**[4]	**Synchorized Visits**	**Cooperation**	**Priority**	**Overtime**	**Care Continuity**	**Total Route**	**AT Changes**[5]
[Hertz and Lahrichi, 2009]	TS		✓				✓	✓	✓									✓	
[Bennett and Erera, 2011]	HEU		✓	✓			✓											✓	
[Nickel et al., 2012]	ALNS		✓	✓	✓		✓	✓	✓							✓	✓	✓	
[Cappanera and Scutellà, 2013]	MILP	✓					✓	✓	✓									✓	
[Mankowska et al., 2014]	aVNS	✓		✓				✓	✓				✓					✓	
[Di Mascolo et al., 2014]	MILP	✓		✓				✓					✓						
[Rest and Hirsch, 2015]	TS	✓		✓				✓	✓	✓						✓		✓	
[Bowers et al., 2015]	HEU	✓					✓	✓							✓		✓	✓	
[Fikar and Hirsch, 2015]	HEU	✓		✓				✓	✓				✓					✓	
[Yuan et al., 2015]	B&P	✓		✓				✓	✓			✓	✓					✓	
[Manerba and Mansini, 2016]	B&P	✓			✓			✓	✓						✓				
[Yalçındağ et al., 2016]	2-PHA	✓		✓			✓	✓	✓									✓	
[Rest and Hirsch, 2016]	TS	✓		✓				✓	✓	✓						✓		✓	
[Heching and Hooker, 2016]	DEC	✓		✓			✓	✓	✓									✓	
[Redjem and Marcon, 2016]	HEU	✓		✓			✓	✓					✓				✓	✓	
[Mankowska, 2016]	DEC	✓		✓				✓	✓				✓					✓	
[Ly Nguyen and Montemanni, 2016]	GA, MILP	✓		✓	✓		✓	✓	✓							✓		✓	
[Fikar et al., 2016]	HEU		✓	✓				✓	✓				✓	✓				✓	
[López-Santana et al., 2016]	MILP		✓	✓		✓	✓	✓	✓					✓	✓			✓	
[Liu et al., 2017]	B&P	✓		✓	✓			✓									✓	✓	
[Yuan and Jiang, 2017]	TS	✓	✓	✓	✓	✓		✓	✓					✓			✓	✓	✓
[Yalçındağ and Matta, 2017]	DEC, MILP	✓		✓			✓	✓									✓		
[Decerle et al., 2017]	MEM	✓		✓				✓	✓				✓					✓	
[Xie and Wang, 2017]	PRA		✓	✓				✓	✓									✓	✓
[Riazi et al., 2017]	B&P, CG	✓		✓				✓	✓									✓	
[Guericke and Suhl, 2017]	ALNS	✓		✓			✓	✓	✓							✓		✓	
[Yuan et al., 2018]	B&P	✓		✓	✓			✓			✓	✓						✓	
[Decerle et al., 2018a]	GA	✓		✓				✓	✓				✓					✓	
[Fikar and Hirsch, 2018]	MAT HEU	✓		✓				✓	✓				✓					✓	
[Decerle et al., 2018b]	MDLS	✓		✓	✓			✓	✓				✓					✓	
[Xiao et al., 2018]	MILP	✓		✓	✓			✓	✓				✓					✓	
[Mascolo et al., 2018]	MILP	✓		✓				✓	✓				✓						

[1]Multi Healthcare Service Provider

[2]Time-dependent Travel Time

[3]Uncertain Travel Time

[4]Uncertain Service Time

[5]Arrival Time Changes

Table 2.2: Previous Survey Comparison - Con't

		Type		Features											Goals					
Author(s)	**Method**	**Static**	**Dynamic**	**Time Windows**	**Unvisited**	**Emergent**	**Multiperiod**	**Multi HSP**	**Multi skills**	**Time-dependent TT**	**Uncertain TT**	**Uncertain ST**	**Synchorized Visits**	**Cooperation**	**Priority**	**Overtime**	**Care Continuity**	**Total Route**	**AT Changes**
[Martinez et al., 2018]	HEU, MILP	✓		✓			✓	✓	✓								✓	✓	
[Nasir and Dang, 2018]	VNS	✓		✓	✓		✓	✓	✓					✓				✓	
[He and Shi, 2019]	SA	✓			✓			✓			✓	✓			✓				
[Martinez et al., 2019]	DEC	✓					✓	✓										✓	
[Ouertani et al., 2019]	GA		✓	✓			✓	✓						✓				✓	
[Liu et al., 2019a]	ALNS	✓		✓				✓					✓					✓	
[Shi et al., 2019]	SA	✓		✓			✓	✓	✓		✓	✓						✓	
[Heching et al., 2019]	DEC	✓		✓	✓		✓	✓	✓										
[Demirbilek et al., 2019]	HEU		✓		✓	✓	✓										✓	✓	
[Hassen et al., 2019]	ArIm		✓	✓	✓	✓		✓	✓					✓	✓		✓	✓	
[Cinar et al., 2021]	ALNS	✓			✓		✓	✓							✓	✓		✓	
[Du et al., 2019]	MA		✓	✓		✓	✓	✓										✓	
[Moussavi et al., 2019]	MAT HEU	✓		✓			✓	✓										✓	
[Ait Haddadene et al., 2019]	NSGA	✓		✓				✓	✓				✓					✓	
[Liu et al., 2019b]	B&P	✓		✓	✓			✓	✓		✓	✓						✓	
[Grenouilleau et al., 2019]	HEU	✓		✓	✓		✓	✓	✓	✓						✓	✓	✓	
[Riazi et al., 2019]	CG	✓		✓				✓	✓									✓	
[Decerle et al., 2019]	AC	✓		✓				✓	✓				✓					✓	
[Shahnejat-Bushehri et al., 2019]	SA, TS	✓		✓			✓	✓	✓				✓					✓	
[Chaieb et al., 2020]	GA	✓		✓	✓			✓	✓						✓			✓	
[Hosseinpour-Sarkarizi et al., 2020]	FGP	✓		✓	✓	✓		✓	✓		✓	✓							
[Dengiz et al., 2020]	MILP	✓		✓				✓	✓									✓	
[Liu et al., 2021]	HEU, MILP	✓		✓				✓	✓				✓					✓	
[Fathollahi-Fard et al., 2021]	HEU	✓		✓	✓		✓	✓	✓							✓	✓	✓	
[Demirbilek et al., 2021]	HEU GRE		✓		✓	✓	✓		✓		✓							✓	
[Malagodi et al., 2021]	MILP, DEC	✓					✓	✓	✓							✓	✓	✓	
Our Study	LGP, GRASP	✓	✓		✓	✓		✓		✓				✓	✓	✓	✓	✓	✓

The HHCRSPs differ in terms of the number of periods in the planning horizon, as well as HSP skills (whether they are multi or not). If a single-day plan is generated as in our study, we classify those studies as single-period [Mankowska, 2016, Di Mascolo et al., 2014, Rest and Hirsch, 2015, Fikar and Hirsch, 2015, Yuan et al., 2015, Manerba and Mansini, 2016, Rest and Hirsch, 2016, Heching and Hooker, 2016, Redjem and Marcon, 2016, Mankowska, 2016, Fikar et al., 2016, Liu et al., 2019b, Yuan and Jiang, 2017, Decerle et al., 2017, Xie and Wang, 2017, Riazi et al., 2017, Yuan et al., 2018, Decerle et al., 2018b, Decerle et al., 2018a, Fikar and Hirsch, 2018, Xiao et al., 2018, Mascolo et al., 2018, He and Shi, 2019, Demirbilek et al., 2019, Demirbilek et al., 2021]. The planning horizon is more than one day, we classify those studies as multi-period [Hertz and Lahrichi, 2009, Bennett and Erera, 2011, Nickel et al., 2012, Cappanera and Scutellà, 2013, Bowers et al., 2015, Yalçındağ et al., 2016, Heching and Hooker, 2016, Redjem and Marcon, 2016, Ly Nguyen and Montemanni, 2016, López-Santana et al., 2016, Yalçındağ and Matta, 2017, Guericke and Suhl, 2017, Martinez et al., 2018, Nasir and Dang, 2018, Martinez et al., 2019, Ouertani et al., 2019, Shi et al., 2019, Heching et al., 2019, Demirbilek et al., 2019, Cinar et al., 2021, Du et al., 2019, Moussavi et al., 2019, Grenouilleau et al., 2019, Shahnejat-Bushehri et al., 2019, Fathollahi-Fard et al., 2021, Demirbilek et al., 2021, Malagodi et al., 2021]. Studies may also be classified based on the number of HSPs. In [Bennett and Erera, 2011, Demirbilek et al., 2019, Demirbilek et al., 2021] studies, they solved home healthcare routing problem while the dynamically arriving new patients for the single HSP. In the multi-HSP feature, which is mentioned as future work in the single-HSP articles, the route of more than one HSP is optimized in the problem set at the same time. Synchronization feature and cooperation strategy can also be used in multi-HSPs. If a patient needs to be treated by multi-HSPs at the same time, the correct timing there is called synchronization in the HHC literature. Differently, cooperation can be defined as a method that aims to reach more patients by dynamically sharing patients among multi-HSPs. Situations where synchronization requires multi-HSPs in the treatment of a patient were studied [Mankowska et al., 2014, Di Mascolo et al., 2014, Fikar and Hirsch, 2015, Yuan et al., 2015, Redjem and Marcon, 2016, Mankowska, 2016, Fikar et al., 2016, Decerle et al., 2017, Decerle et al., 2018b, Decerle et al., 2018a, Fikar and Hirsch, 2018, Xiao et al., 2018, Mascolo et al., 2018, Liu et al., 2019b, Ait Haddadene et al., 2019, Decerle et al., 2019, Shahnejat-Bushehri et al., 2019, Liu et al., 2021]. The cooperation strategy [Fikar et al., 2016, López-Santana et al., 2016, Yuan and Jiang, 2017, Nasir and Dang, 2018, Ouertani et al., 2019, Demirbilek et al., 2019, Hassen et al., 2019, Du et al., 2019, Fathollahi-Fard et al., 2021] as less common in the literature aims to handle the new urgent patients. That can be thought as a kind of solidarity. More patients or higher collected priority value can be reached with the cooperation strategy.

Several researchers have focused on travel time and service time uncertainties. Mostly, travel time and service time are considered to be uncertain simultaneously, but Demirbilek, M. et al. [Demirbilek et al., 2021] studied only the case of uncertain travel times. Uncertainties in service time arise due to the health status of the patient, the level of experience of the caregiver and the difficulty of the procedures. Uncertainties in travel time are also caused by traffic and road conditions. Also, Yuan, B. et al. [Yuan et al., 2015] worked as uncertain, with both travel and service times. They studied daily scheduling of caregivers with stochastic times. This paper mainly focused on uncertain parameters like traffic, health conditions of patients. This study aimed to under control the uncertainty with stochastic model. They worked on the chance constraint which refers to whether the patient can be visited on the time or not. The chance constraint guaranteed the service quality because the chance factor is limited by a threshold. The authors choose to branch-and-price(B&P) algorithm to work on this issue. The algorithm mainly focused on reducing cost of travel, failure and recompense of uncovered patients.

Another rarely-addressed feature is time-dependent travel times. In several studies travel times between locations vary by time [Rest and Hirsch, 2015, Rest and Hirsch, 2016, Grenouilleau et al., 2019]. Time-dependent travel times are mainly caused by the departure times of public transportation and the traffic density changes throughout the day. In addition to this feature, Rest, Klaus-Dieter and Hirsch, Patrick [Rest and Hirsch, 2015, Rest and Hirsch, 2016] worked on multiple transportation modes, such as walking and public transportation, with time-dependent travel times.

In terms of objective functions, a trend that goes from single objective [Hertz and Lahrichi, 2009, Bennett and Erera, 2011, Cappanera and Scutellà, 2013, Mankowska et al., 2014, Di Mascolo et al., 2014] to multi-objective approaches from past to present is observed. Except Di Mascolo, M. et al. which focused on minimizing the total waiting time, all single-objective papers focused on minimizing the total route distance or the total travel time [Di Mascolo et al., 2014]. When considering multiple objectives, improvement is generally aimed from the perspective of three main entities: patients, HSPs, and HHC managers. Since the objectives for patients are approached from the perspective of customer satisfaction, patient-oriented objectives could be classified as the following: minimizing the delay in time windows [Decerle et al., 2019, Decerle et al., 2018a, Mascolo et al., 2018, Decerle et al., 2017, Decerle et al., 2018b, Nickel et al., 2012], minimizing waiting times [Di Mascolo et al., 2014, Rest and Hirsch, 2015, Rest and Hirsch, 2016, Di Mascolo et al., 2014, Martinez et al., 2019, Guericke and Suhl, 2017, Mankowska et al., 2014, Chaieb et al., 2020], meeting their needs and preferences [Malagodi et al., 2021, Bowers et al., 2015, Ait Haddadene et al., 2019, Mascolo et al., 2018, Hassen et al., 2019, Hosseinpour-Sarkarizi et al., 2020, Redjem and Marcon, 2016], and maintaining continuity of care [Nickel et al., 2012, Bowers et al., 2015, Redjem and Marcon, 2016, Liu et al., 2017, Yuan and Jiang, 2017, Yalçındağ and Matta, 2017, Martinez et al., 2018, Demirbilek et al., 2019, Hassen et al., 2019, Grenouilleau et al., 2019, Fathollahi-Fard et al., 2021, Malagodi et al., 2021].

In dynamic problems, minimizing the changes in patient visit times is one of the objectives and also, causing minimum change in the routes and not doing overtime by HSPs are taken into account. On the other hand, HHC managers generally have cost-oriented goals, including total route minimization, overtime minimization, and the use of a minimum number of outsource practices. Studies in the literature that address multi-objective models have mostly taken the objective scalarization approach [Fathollahi-Fard et al., 2021, Decerle et al., 2017, López-Santana et al., 2016, Dengiz et al., 2020, Mascolo et al., 2018], in addition to developing multi-criteria heuristics [Hassen et al., 2019] such as NSGA, that generate Pareto optimal solutions [Ait Haddadene et al., 2019], or multi-directional local search [Decerle et al., 2018b], and goal programming models [Hosseinpour-Sarkarizi et al., 2020].

In terms of solution methods, many different integer programming approaches have been implemented. Most studies present a static MILP model [Mankowska et al., 2014, Di Mascolo et al., 2014, López-Santana et al., 2016, Xiao et al., 2018, Mascolo et al., 2018, Dengiz et al., 2020], and solution techniques such as branch and price [Yuan et al., 2015, Manerba and Mansini, 2016, Liu et al., 2017, Yuan et al., 2018, Liu et al., 2019b], column generation [Riazi et al., 2017, Riazi et al., 2019], Bender's decomposition [Heching and Hooker, 2016, Mankowska, 2016, Yalçındağ and Matta, 2017, Martinez et al., 2019, Heching et al., 2019], and heuristic algorithms are usually preferred. Adaptive large neighborhood search (ALNS) [Nickel et al., 2012, Guericke and Suhl, 2017, Liu et al., 2019a, Cinar et al., 2021], variable neighborhood search (VNS) [Mankowska et al., 2014, Nasir and Dang, 2018], Tabu search [Hertz and Lahrichi, 2009, Rest and Hirsch, 2015, Rest and Hirsch, 2016, Yuan and Jiang, 2017, Shahnejat-Bushehri et al., 2019], genetic algorithm [Ly Nguyen and Montemanni, 2016, Decerle et al., 2018b, Ouertani et al., 2019, Chaieb et al., 2020], simulated annealing [He and Shi, 2019, Shi et al., 2019], and ant colony optimization algorithm [Decerle et al., 2019] are the mostly resorted heuristics. In addition, some hybrid algorithms have also been provided [Lin et al., 2021, Ly Nguyen and Montemanni, 2016, Martinez et al., 2018]. Fuzzy goal programming was proposed by Hosseinpour-Sarkarizi et al. [Hosseinpour-Sarkarizi et al., 2020]. Thus, a wide range of methods have been covered in the existing studies.

We believe comparing the cost of healthcare services and the value of caring for the patients is not meaningful. Therefore, in our study, we used the Lexicographic Goal Programming (LGP) method, differing from the literature. The LGP is used when the objective functions have clear supremacy over each other. Also, the LGP has compatibility with real-life practices and takes care of the difficulties in scalarizing different types of objectives. For the static problem, we preferred the GRASP + VNS metaheuristic due to the need for a short run time. GRASP + VNS heuristic has been successfully implemented in vehicle routing problems, which is a special case of our problem (we have a selective routing problem). For the dynamic problem, we preferred to use the GRASP + VNS metaheuristic hybridizing it with MILP, since the size of the problem decreases as the day progresses and MILP becomes tractable. The MILP is solved first with a run time limit, and if it returns the optimal solution within a reasonable run time, this solution is used. If it cannot, a solution is obtained through the metaheuristics, and then the better one is used.

2.2 Contributions to the Literature

Our study contributes to the existing literature mainly by addressing a combination of key problem features to represent a realistic setting, and in particular, by proposing a cooperative approach while ensuring care continuity and incorporating time-dependent travel times for the first time in the literature. We specifically incorporate the cooperation of the HSPs in a dynamic setting and time-dependent travel times into the problem. We consider the following objectives simultaneously and aim to generate solutions that excel in these objectives according to their importance rankings defined in a health care setting: i) maximization of visited patient priorities, ii) minimization of the arrival time changes, iii) minimization of the overtime, iv) minimization of the total travel time. Moreover, we also impose a deadline for servicing emergent patients.

In terms of solution methodology, we present a fast and effective heuristic (GRASP + VNS metaheuristic) designed in accordance with lexicographic goal programming. In our computational study, we quantify the benefit of HSPs' cooperation, especially in cities where traffic is a problem, through a case study where the increasing daily demand to home healthcare service lead to prioritization of the patients, together with emergent patients that necessitate re-optimization of the schedule. We believe that the addressed problem poses a setting applicable in practice and extends to emergency situations.

Chapter 3

PROBLEM DEFINITION

Home healthcare routing and scheduling problems have been studied in the literature with different variants, as discussed in Chapter 2. In this chapter, we address two related problems in this field. The first is a static one solved at the beginning of the day to generate an initial schedule for a single HSP. The second one is solved each time an emergent call occurs during the day, accounting for the cooperation between multiple HSPs. Thus, we call this problem the dynamic problem.

3.1 Static Problem

In the static problem, two different objectives exist: the maximization of the total priorities of visited patients, represented by total collected priority from the visits, and the minimization of the route duration. The collected priorities represent the priorities assigned to the patients according to features such as patient's age and health status. Lexicographic goal programming approach is adopted, where maximizing the total priority has the first priority, followed by minimization of the route duration, since servicing the most number of the "right" patients carries more importance.

The static problem imposes that a patient can be visited at most once in a day, and the route must be completed within the working hours. Patient visits that cannot be included in the daily plan at the beginning of the day are postponed to the following days.

3.2 Dynamic Problem

The dynamic problem departs from the literature as it manages urgent requirements arising throughout the day by allowing the HSPs to visit patients outside their patient list for better utilization of the overall service capacity. Hence, the problem coordinates multiple HSPs. Re-optimization is triggered as soon as an emergent patient call comes. HSPs can be in two different states when the call arrives; either moving from one patient to another or treating a patient. Let us consider an HSP servicing a patient. When an urgent patient call is received, she/he cannot leave the patient immediately and will follow the new route after the treatment is over. The location of the patient, called the current location of the HSP, is the starting node of that HSP in the new route. We refer to the end time of the service as the first available time. In the case where the HSP is on the road, the position of the last visited patient is accepted as the current location. In both these situations of HSP, we called the current locations the "frozen locations". The first available time is set to the moment the emergent call comes. Patient visits postponed after re-optimization are re-evaluated for scheduling each time a new patient visit request arrives. It is also supported by Figure below.

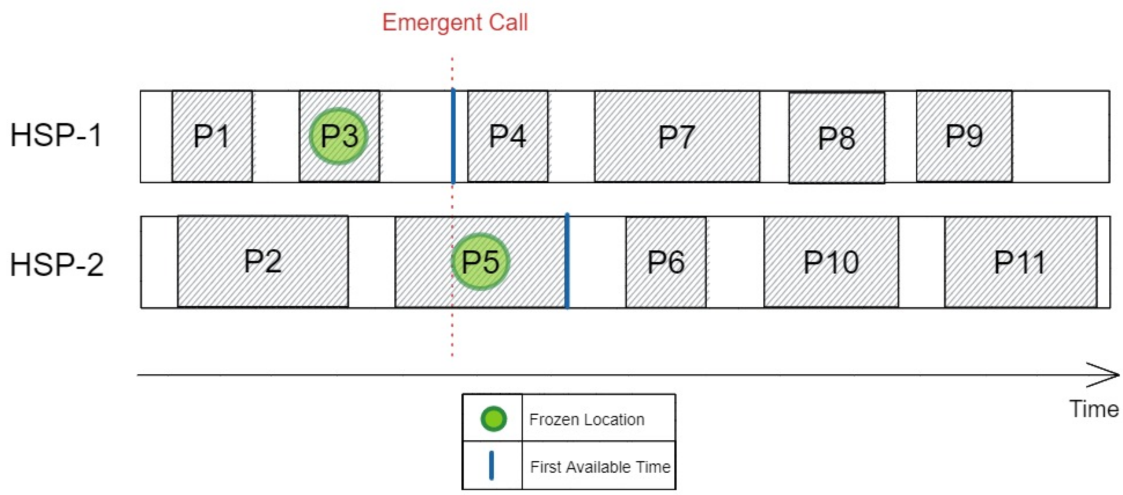


Figure 3.1: An illustrative example for the frozen node and the first available time.

Figure 3.1 gives the schedule that HSPs will follow during the day. Since the HSP1 is on its way from P3 to P4 when the emergent call comes in, the time of the emergent call is taken as the first available time of the HSP1. However, since we cannot access the real time location information, the frozen node of HSP1, that is, the current location that we will use in re-optimization will be the location of patient

P3. Since the HSP2 patient is visiting P5 when the emergent call arrives, the first available time of the HSP2 is taken as the end time of P5's visit. The location of P5 is taken as the frozen node of HSP2.

The objective functions of the dynamic problem include the minimization of arrival time changes in the schedule and minimization of total overtime. The changes in the arrival times are minimized for the sake of avoiding changes in patients' daily plans and reducing the negative effect of frequently changing the routes of the HSPs; thus keeping patient and HSP satisfaction high. The minimization of overtime has been taken into account because the costs are higher in overtime compared to regular shifts and excessive overtime impacts the life quality of HSPs negatively. Also, in a pandemic situation, overtime increases infection risk. Our other objectives are maximization of priority collected and minimization of the total time of routes, as in the static problem. Lexicographic goal programming (LGP) is adopted here as well since the objectives have clear dominance over each other. The highest to lowest lexicographic order entails maximizing the total priority collected, minimizing the total arrival time changes, minimizing the total overtime, and finally minimizing the sum of route duration. This order is chosen to prioritize services to the most urgent patients. Then, the visit time changes, that is, going any earlier or later than the previously informed time is avoided among the solutions that do not compromise from the first target values. Among the solutions that can be obtained without giving up on the first two goals mentioned, the solution that minimizes total overtime is preferred. Next, the solution whose total travel time is minimized without compromising from the other objectives is selected.

Each HSP is responsible from a specific patient group. To enable a patient to be serviced by the same HSP over time, i.e., to achieve care continuity, we give higher priorities to patient-HSP pairs where the patient belongs to the HSP's group, and lower priorities to pairs such that the patient is not in the group of that HSP. For example, suppose P_1 is a patient regularly treated by $HSP1$. Since $HSP1$ is more knowledgeable about the patient's history and the last treatment applied, the priority value for P_1 will be higher for $HSP1$ compared to values for other HSPs. This is, of course, valid under conditions where the patient is satisfied with his or her service. In other words, customer satisfaction is also handled by priority values. As expected, the primary component of a priority value is the patient's health status. If the patient's health condition is severe, regardless of the HSP, the highest priority is assigned. Also, emergent calls possess the highest priority for all HSPs.

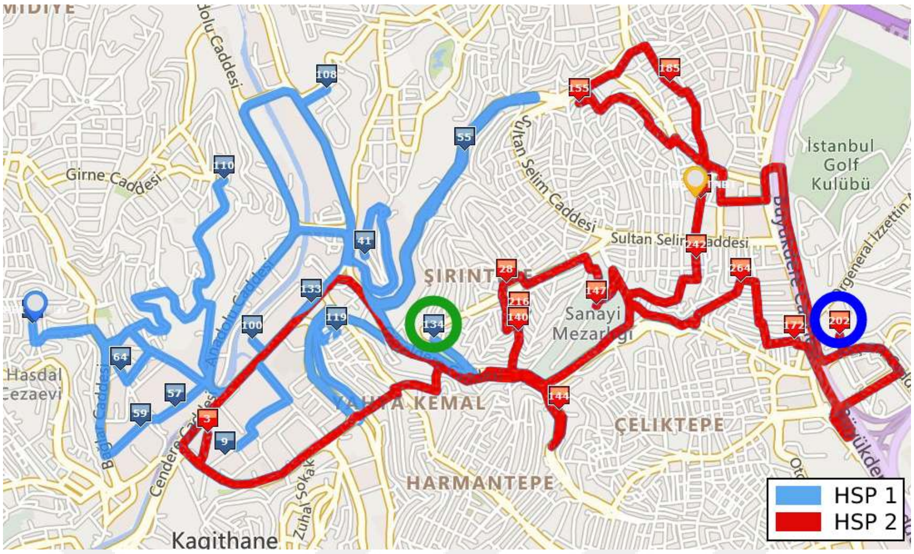
A patient is informed the arrival time whenever the initial or the re-optimized schedule is generated. The last arrival time, the o_i parameter, conveyed to patient i is input to the problem and the absolute difference between the old arrival time and the new arrival time is kept by the c_i parameter, whose sum over the patients is minimized. Furthermore, the o_i parameters of the emergent patient calls are marked as the call time to keep track of the waiting time of emergent patients. Then, the maximum allowed waiting time for emergent patients is defined as two hours.

In densely populated cities, traffic conditions change throughout the day. Therefore, we use estimated travel times in different time intervals. Today's navigation systems, such as Google Maps, YandexMaps, and BingMaps, perform these estimations successfully. Time-varying travel times between patient locations taken from *Microsoft Bing Maps* that are estimated according to traffic density are inputs for both of the static and dynamic problems.

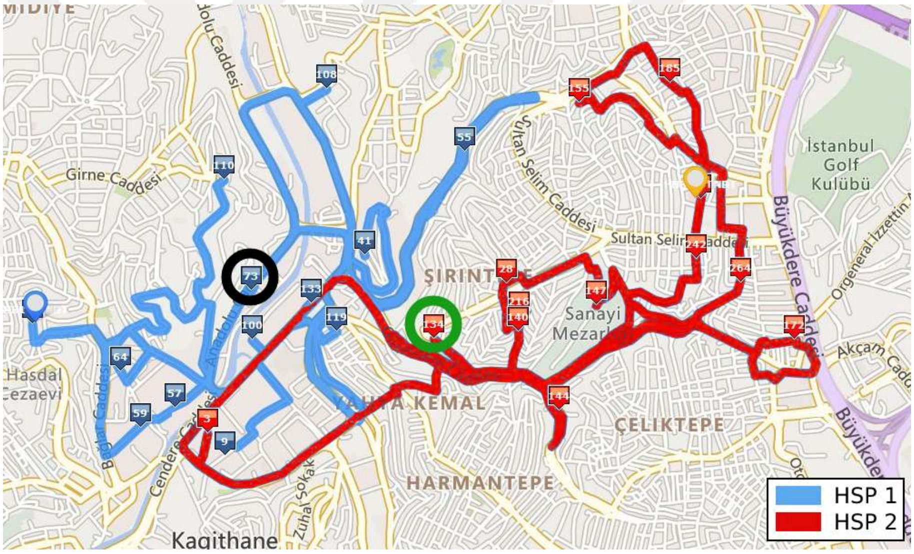
The necessity to handle urgent requests that come dynamically during the day may cause infeasibility. In order to overcome this, overtime and patient visit postponements are made, even if undesirable. A maximum overtime is set as four hours in our study. The postponements are penalized both by failing to collect priority in the first objective and by giving the arrival time change value as a one-day working hour duration in the second objective.

The inclusion of an emergent patient on the route is shown in Figure 3.2. Nodes in blue represent patients on HSP1's visit list. Nodes in red represent patients that HSP2 will visit.

Four situations can occur when an emergent patient is included in the route. First of all, the patients to be visited on the routes remain the same and the emergent patient is added to the route. The second situation, the patients remain the same, but with the cooperation strategy, the HSP, which will take care of some patients, can be changed. The third situation may be those who are postponed from the patients.



(a)



(b)

Figure 3.2: An illustrative example showing the change in the routes of HSP1 and HSP2 with the arrival of an emergent patient: (a) the route before the arrival of the emergent patient, (b) the route after the arrival of the emergent patient

As the final situation, some patients are postponed and some of the patients' HSP is changed with the cooperation strategy.

This study focuses on generating daily plans, which can be incorporated into long term planning by updating the priorities of unvisited patients over the days.

Therefore, we consider that the priorities of the patients will be increased day by day if they are postponed, as mentioned in the study [Cinar et al., 2021]. The daily emergent patients have negative effects on long-term planning. The postponed patients will increase the size of the patient set for the next days' plannings. In our problem, we aim to reduce this postponement problem using the cooperation strategy. Also, since the priority values of the postponed patients will increase over the days, they will have more probability of being included in planning.

For the sake of clarity, we have created a figure of an output that is an example of the final situation. Routes of sample "20-High-0" are shown in Figure in 3.2. *Patient-75* figure came to the routes in 3.2a as emergent patient and then 3.2b routes were created. The location of *Patient-75* on the map is indicated by the black circle. When *Patient-75* was included in the route, *Patient-134* also switched from HSP1 to HSP2's route. *Patient-134* is indicated by a green circle on the map. Also, *Patient-202* was postponed while in HSP2's route. *Patient-134* is indicated by a red circle on the map. Maps were pulled from Microsoft Bing with API, where we also created our travel distance matrix.

When Figure 3.2 is examined, the difference of using a real map can be understood. For example, although the euclidean distance between *Patient-55* and *Patient 108* is short, the actual travel time will be long because there is no direct path between them.

Chapter 4

METHODOLOGY

In this chapter, we present the MILP formulations of both the static and the dynamic problems, the proposed GRASP+VNS metaheuristic for initial optimization, the GRASP-VNS meta-heuristic for re-optimization, and the framework of the hybrid re-optimization procedure.

For the dynamic problem, we proposed a re-optimization procedure for all approaches. In this re-optimization procedure, the MILP models, GRASP+VNS metaheuristic algorithms, and the Hybrid approach try to solve the problem set again with the emergent patients when a new emergent patient arrives. An illustrative flow chart is shown in Figure 4.1.

In Figure 4.2, the LGP steps are shown. Firstly, the MILP model/GRASP+VNS metaheuristic algorithm is solved for Goal 1. If the MILP model is solved, the goal value is added as a constraint for the next goal solution. The constraint guarantees that the next goal solution cannot have a worse goal value for previous goals. If the GRASP+VNS metaheuristic algorithm is solved, the goal values are saved for the next goal solutions. These saved values are compared with the solutions for the next goals. If the next solutions cannot get better goal values, the current solution remains the best solution. In the MILP models, objective functions are changed by goals. In the GRASP+VNS algorithms, the candidate scores of the solutions are calculated according to the current goal. The candidate scores show the quality of the solutions for the current goal. These steps continue until the solutions are reached for the last goal in the LGP. The same approach is used for both static and dynamic problems.

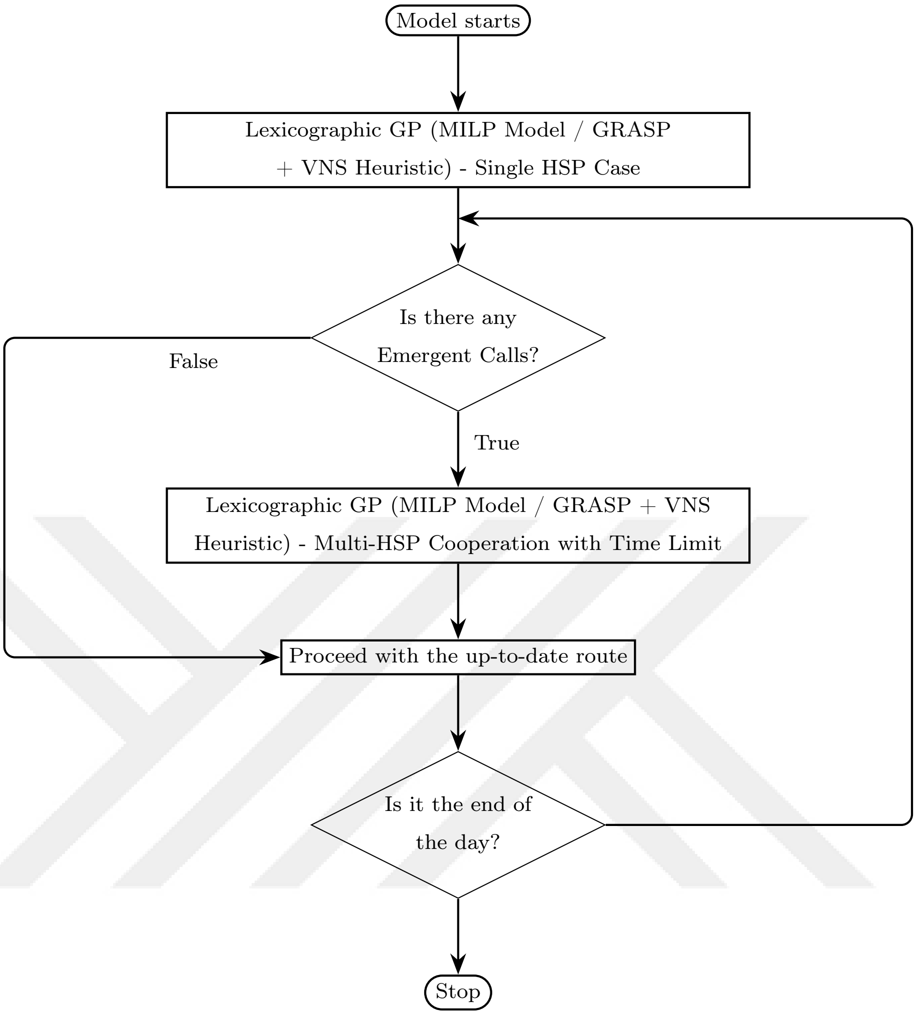


Figure 4.1: The re-optimization flow for a single day

4.1 MILP Models

In our first model, the route to be followed initially is determined for a single HSP, and in the second model, a daily route for each HSP is created, considering all HSPs. Multiple objectives are defined as explained in the previous section. It should be noted that these models are not substitutes for each other but complement each other. The initial routes of all HSPs are inputs to the dynamic problem. In the dynamic problem emergent arrivals have strict time windows. The two MILP formulations are explained below.

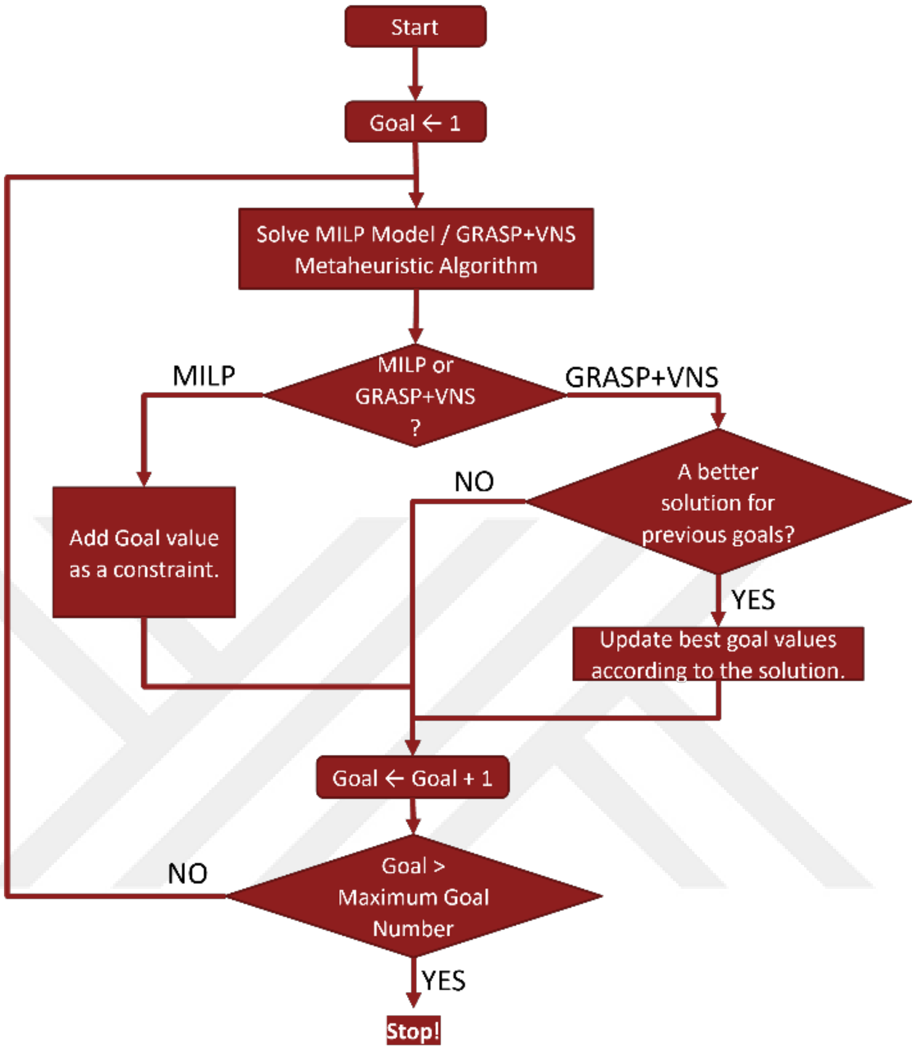


Figure 4.2: The flow chart of the Lexicographic Goal Programming (LGP) Procedure.

The MILP Model of the Static Problem: Routing and Scheduling for a Single HSP

The MILP model generates a single day scheduling and routing plan for one HSP. The output of this model is used in re-optimization as input. The MILP model is first solved to maximize the total priority of patients selected to be visited. After the solution is obtained, the model is solved again to minimize the total overtime of the HSP, but this time, the objective function value of the first goal is maintained by means of a constraint. The MILP finds the same total priority with possibly different patients in the second solution.

The travel time between two patients is calculated according to which time interval it is associated with. At the end of the day, the HSP is expected to return to the starting point. As stated before, patients in the group of a particular HSP are assigned a high priority value for this HSP to ensure continuity of care.

The MILP Model Formulation for the Static Problem

The below table shows the input parameters and decision variables for MILP model of the single HSP problem.

Table 4.1: Sets, parameters and decision variables

Sets	
$\{0\} \cup P$	The set of patients is $P = 1, ..., n\}$. The service provider's home base is indexed as $\{0\}$.
$P^{'}$	The set of patients with the highest priority value which is 5.
T	The set of time blocks.

Parameters	
p_i	The priority of patient i; higher p_i values belong to more prioritized patients.
s_i	The service time required to serve patient i.
e_k	The start time of the time block k.
l_k	The finish time of the time block k.
t_{ijk}	The time required to travel between locations of patients i and j in k time block.
E	The day beginning time of service providers.
W	The day ending time of service providers.
M	A sufficiently big number.

Decision Variables	
x_{ijk}	Binary variable indicating whether patient j is visited immediately after patient i in the time block k.
y_{ik}	Binary variable indicating whether patient i is visited in the time block k.
a_i	The arrival time to the location of patient i.

$$z^1 = \max \sum_{i \in P} \sum_{k \in T} p_i y_{ik}$$

$$z^2 = \min \sum_{i \in P} \sum_{k \in T} s_i y_{ik} + \sum_{i \in P \cup \{0\}} \sum_{j \in P \cup \{0\}} \sum_{k \in T} t_{ijk} x_{ijk}$$

s.t.

$$\sum_{\substack{j \in P \cup \{0\} \\ j \neq i}} x_{ijk} = y_{ik} \qquad \forall i \in P, k \in T \qquad (1)$$

$$\sum_{\substack{j \in P \cup \{0\} \\ j \neq i}} x_{jik} = y_{ik} \qquad \forall i \in P, k \in T \qquad (2)$$

$$\sum_{k \in T} y_{ik} \leq 1 \qquad \forall i \in P \qquad (3)$$

$$\sum_{j \in P} x_{0j1} = 1 \qquad (4)$$

$$\sum_{k \in T} \sum_{i \in P} x_{i0k} = 1 \qquad (5)$$

$$\sum_{k \in T} \sum_{i \in P} s_i y_{ik} + \sum_{k \in T} \sum_{i \in P} \sum_{j \in P} t_{ijk} x_{ijk} \leq W - E \qquad (6)$$

$$a_i + s_i + t_{ijk} - a_j \leq M(1 - x_{ijk}) \qquad \forall i, j \in P, i \neq j, k \in T \qquad (7)$$

$$a_i \geq \sum_{k \in T} y_{ik} e_k \qquad \forall i \in P \qquad (8)$$

$$a_i \leq M(1 - \sum_{k \in T} y_{ik}) + \sum_{k \in T} y_{ik} l_k \qquad \forall i \in P \qquad (9)$$

$$a_i \geq \sum_{k \in T} x_{i0k} e_k \qquad \forall i \in P \qquad (10)$$

$$a_i \leq M(1 - \sum_{k \in T} x_{i0k}) + \sum_{k \in T} x_{i0k} l_k \qquad \forall i \in P \qquad (11)$$

$$a_i \geq E + x_{0i1} t_{0i1} \qquad \forall i \in P \qquad (12)$$

$$\sum_{k \in T} y_{ik} = 1 \qquad \forall i \in P^{'} \qquad (13)$$

$$\sum_{k \in T} x_{iik} = 0 \qquad \forall i \in P \qquad (14)$$

$$E \leq a_i \leq W \qquad \forall i \in P \qquad (15)$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j \in P \cup \{0\}, k \in T \qquad (16)$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in P, k \in T \qquad (17)$$

Apart from the two objectives defined before, constraints (1)-(2) ensure that if a patient is visited, then either the patient is the first patient to be visited or there is another patient who was visited before him. Similarly, the patient is either the last patient to be visited or there will be another patient who is going to be visited right after him. Constraint (3) represents whether a patient will be visited in any time interval.

Constraints (4)-(5) enforce that the service provider has to start from a starting location and return to the same location at the end of the day. Constraint (6) guarantees that the total time of the route cannot exceed the defined working hours. Constraints (7) make sure that if patient j is visited immediately after patient i in time block k, then the time to begin service for patient j must be at least $s_i + t_{ijk}$ after the HSP arrives at patient i. These constraints also prevent sub-tours. Constraints (8)-(9) impose that if a patient is selected to be visited, then the arrival time has to be within the time block. Constraints (10)-(11) are similar to constraints (8)-(9) and are defined for the start and end points. Constraints (12) state that the first patient in the route should be visited after a duration including the travel time from the starting point to the patient location in the first time block. There is no waiting time for the HSP before beginning to work. Constraints (13) enforce that the highest-prioritized patients must be visited in the route. Constraints (14) impose that the HSP should go from one patient to either another patient or the starting point. Constraints (15)-(17) define domains of the decision variables.

After the model is solved with the first objective, the optimal $(z^1)^*$ value is used in the second model in the following constraint:

$$\sum_{i \in P} \sum_{k \in T} p_i y_{ik} = (z^1)^*$$

The MILP Model of the Dynamic Routing and Scheduling Problem with Multi-HSP Cooperation

The routes given to the service providers at the beginning of the day are initially taken as input. The model imposes that the patients with the highest priority must be visited. The priorities of the newly arrived patients are also at the highest level. If a patient has the highest priority, it does not matter who will care for him. The healthcare workers may exchange patients in order to reach a better solution or to visit patients who are required to be visited. Unlike the first model, overtime is allowed. In re-optimization, the current visit times of newly arrived patients are kept constant. Also, emergent calls must be visited in two hours. If a patient is not visited, the change in arrival time is considered as one day; in this way, postponement is avoided.

The model is solved each time a new emergent call arrives. At that moment, the locations of the service providers are found in addition to when they will be available and given to the model as input. We refer to the locations where they are as frozen locations. Previous visits, including their frozen locations, are marked as visited. Planned visits (not visited yet), and emergent calls are inputs. The model is solved in a lexicographic fashion with the constraints and objectives described before. As the output of the model, new patients are added to the routes, the service providers that will visit the patients are updated, the postponed patients are determined, the arrival times to the patients are updated, and the overtime information is updated.

The MILP Model Formulation of the Dynamic Problem with Emergent Patients and Multi-HSP Cooperation

There are four lexicographic goals, in priority order: (i) maximizing the total priority of visited patients, (ii) minimizing the arrival time changes, (iii) minimizing the total overtime, and (iv) minimizing the total time of the routes. We note that goals (ii) and (iii) do not exist in the static problem.

The list in Table 4.2 shows the input parameters and decision variables of the model.

Table 4.2: Sets, parameters and decision variables

Sets	
N	The set of healthcare service providers.
P	The set of patients is $P = \{n, ..., n + p - 1\}$, where n is the number of healthcare service providers.
P^e	The set of emergent patients.
$P^{'}$	The set of patients with the highest priority value which is 5. $\mathrm{P}^{'} \supseteq P^e$.
L	The set of nodes is $N \cup P \cup \{n + p, ..., n + p + n - 1\}$, where n is the number of healthcare service providers, and p is the number of patients.
T	The set of time blocks.

Parameters	
p_{in}	The priority of patient i for the healthcare service provider n.
s_i	The service time required to serve patient i.
e_k	The start time of the time block k.
l_k	The finish time of the time block k.
t_{ijk}	The time required to travel between locations of patients i and j in k time block.
q_n	The initial location of the healthcare service provider n. The value is equal to L_{n-1}.
r_n	The returning location of the healthcare service provider n. The value is equal to L_{p+2n-1}.
R	The re-optimization time.
W	The day ending time of service providers.
U	Maximum allowed overtime duration per healthcare service provider.
γ	Maximum allowed waiting time duration per emergent patient.
o_i	The last arrival time information of the location i, where $i \in N \cup P$.
β_n	Avaliable time of the healthcare service provider n. β_n is equal to $max\{o_{q_n} + s_{q_n}, R\}$.
M	A sufficiently big number.

Decision Variables	
x_{ijkn}	Binary variable indicating whether patient j is visited immediately after patient i by the HSP n in the time block k .
y_{in}	Binary variable indicating whether patient i is visited by the HSP n.
a_i	The arrival time to the location of patient i.
u_n	The overtime amount of the HSP n.
c_i	The absolute duration change on arrival time of the patient i.

The formulation is as follows.

$$z^1 = \max \sum_{i \in P} \sum_{n \in N} p_{in} y_{in}$$
$$z^2 = \min \sum_{i \in P} c_i$$
$$z^3 = \min \sum_{n \in N} u_n$$
$$z^4 = \min \sum_{n \in N} a_{r_n}$$

s.t.

$$\sum_{\substack{j \in P \\ j \neq i}} \sum_{k \in T} x_{ijkn} + \sum_{k \in T} x_{ir_nkn} = y_{in} \qquad \forall i \in P, n \in N \qquad (1)$$
$$\sum_{\substack{j \in P \\ j \neq i}} \sum_{k \in T} x_{jikn} + \sum_{k \in T} x_{q_nikn} = y_{in} \qquad \forall i \in P, n \in N \qquad (2)$$
$$\sum_{n \in N} y_{in} \leq 1 \qquad \forall i \in P/P^{'} \qquad (3)$$
$$\sum_{n \in N} y_{in} = 1 \qquad \forall i \in P^{'} \qquad (4)$$
$$\sum_{j \in L/N} x_{q_nj1n} = 1 \qquad \forall n \in N \qquad (5)$$
$$\sum_{i=0}^{n+p-1} \sum_{k \in T} x_{ir_nkn} = 1 \qquad \forall n \in N \qquad (6)$$
$$y_{in} \leq 1 - \sum_{k \in T} x_{q_nr_nkn} \qquad \forall j \in P, n \in N \qquad (7)$$
$$a_i + s_i + t_{ijk} - a_j \leq M(1 - \sum_{n \in N} x_{ijkn}) \qquad \forall i, j \in P, k \in T \qquad (8)$$
$$\sum_{j \in P} \sum_{k \in T} x_{q_njkn}(\beta_n + t_{q_njk}) \leq a_j \qquad \forall j \in P \qquad (9)$$

cont'd - s.t.

$$a_i + s_i + \sum_{k \in T} t_{ir_nk} x_{ir_nkn} \leq a_{r_n} + M(1 - \sum_{k \in T} x_{ir_nkn}) \quad \forall i \in P, n \in N \quad (10)$$

$$\beta_n + \sum_{k \in T} t_{q_nr_nk} x_{q_nr_nkn} \leq a_{r_n} + M(1 - \sum_{k \in T} x_{q_nr_nkn}) \quad \forall n \in N \quad (11)$$

$$a_{r_n} - W \leq u_n \quad \forall n \in N \quad (12)$$

$$a_i + s_i \geq \sum_{k \in T} \sum_{n \in N} e_k x_{ijkn} \quad \forall i \in P, \; j \in P \cup r. \quad (13)$$

$$a_i + s_i \leq \sum_{k \in T} \sum_{n \in N} l_k x_{ijkn} + M(1 - \sum_{k \in T} \sum_{n \in N} x_{ijkn}) \quad \forall i \in P, \; j \in P \cup r. \quad (14)$$

$$a_i - o_i \leq c_i \quad \forall i \in P \quad (15)$$

$$o_i - a_i \leq c_i \quad \forall i \in P \quad (16)$$

$$\sum_{n \in N} y_{in} = 1 \quad \forall i \in P' \quad (17)$$

$$a_i \leq o_i + \gamma \quad \forall i \in P^e \quad (18)$$

$$c_i \geq (W + U)(1 - \sum_{n \in N} y_{in}) \quad \forall i \in P \quad (19)$$

$$\sum_{k \in T} \sum_{n \in N} x_{iikn} = 0 \quad \forall i \in L \quad (20)$$

$$R \leq a_i \leq W + U \quad \forall i \in P \quad (21)$$

$$0 \leq u_n \leq U \quad \forall n \in N \quad (22)$$

$$0 \leq c_i \quad \forall i \in P \quad (23)$$

$$y_{in} \in \{0, 1\} \quad \forall i \in P, n \in N \quad (24)$$

$$x_{ijkn} \in \{0, 1\} \quad \forall i, j \in P, \; k \in T, n \in N \quad (25)$$

Since x and y decision variables also exist in the MILP model for the static problem, we use them with different indexes in the MILP model for the dynamic problem. Moreover, in x decision variables, there sis an extra "n" index for representing the HSPs. y decision variables have the "n" index for the HSPs instead of the "k" index for the time-intervals. In the MILP model of the dynamic problem, the accuracy of the time-interval information is guaranteed with x decision variables in the constraints (13-14). Also, the parameter p for priorities has a new index which is "n". This index refers to the related HSP n and is used to ensure care continuity.

Constraints (1)-(2) describe that at most one patient can be visited after another or the HSP goes to the finish point; similarly, a patient can be visited after at most one patient or after the starting point of each HSP. If the patient is not on any route, there is no arrival or departure from the patient. According to Constraint (3), patients that do not have the highest priority can be visited by at most one HSP. On the other hand, constraint (4) imposes that patients with the highest priority, which includes the emergent calls, must be visited by an HSP. Constraints (5)-(6) state that each HSP has to start from her/his starting location and return to her/his ending location at the end of the route. Constraints (7) represent that when any HSP goes to the ending location from starting location without visiting any patient, we should not assign any patient to this HSP. The highly-prioritized patients must be visited within the day and this is provided by constraints (17).

Constraints (8) enforce that if patient j is visited immediately after patient i, the arrival time to patient j should be equal to the summation of the arrival time to patient i, the service time of patient i, and the travel time between patients i and j. Constraints (9) imply that the arrival time of the first visited patient by each HSP should equal the sum of the first available time of the HSP and the travel duration required to reach the patient. Constraints (10)-(11) represent that the returning times of HSPs are equal to the sum of the last visited patient arrival time, the service time of the patient, and the travel time. The arrival times must be between the re-optimization time and the maximum allowed working hours, according to constraints (21). Constraints (12) calculate the overtime of a HSP. The overtimes are limited by four hours in constraints (22).

Constraints (13)-(14) guarantee that the travel time duration between patient i and patient j or the *returning node* is obtained from the valid time interval. The reason for this is that travel times vary by the traffic density in the time intervals. The valid time interval is the time interval which includes the completion time of patient i's visit by HSP n.

Constraints (15)-(16) calculate the difference between the expected arrival times and realized arrival times. The expected arrival times are updated according to the last calculated route. Also, the expected arrival times of the emergent patients are equal to their call times. Therefore, the difference represents the waiting times for the emergent patients. The emergent calls have to be responded to in γ duration, and constraints (18) guarantee this duration limit. Constraints (19) enforce that the arrival time change value of the patient is equal to the maximum allowed working hours, if the routes of the HSPs do not include him/her.

Constraints (20) imply that HSPs cannot go to a location from the same location in any time block. Constraints (21)-(25) describe the domains and the types of the decision variables.

Since this model is solved with four goals in order, the optimal objective function values $(z^1)^*$, $(z^2)^*$, and $(z^3)^*$ are added to the model cumulatively. To get the optimal values of the objective functions, the model is solved step-by-step with a new goal constraint as defined below.

$$\sum_{i \in P} \sum_{n \in N} p_{in} y_{in} = (z^1)^*$$

$$\sum_{i \in P} c_i = (z^2)^*$$

$$\sum_{n \in N} u_n = (z^3)^*$$

4.2 GRASP+VNS Metaheuristic Algorithms

Since even the static problem with only the priority collection objective is NP-hard and especially in the dynamic case solutions have to be obtained quickly, we develop heuristic algorithms.

We make use of a GRASP algorithm [Resende and Ribeiro, 2005] to generate an initial feasible solution, and then VNS is implemented to improve the solution. A GRASP+VNS metaheuristic is developed for each of the static and dynamic problems. We describe the heuristics in the following subsections.

The flow charts of GRASP and VNS heuristic algorithms are shown in Figure

4.3 and Figure 4.4, respectively. The details of the algorithms are explained in the following subsections.

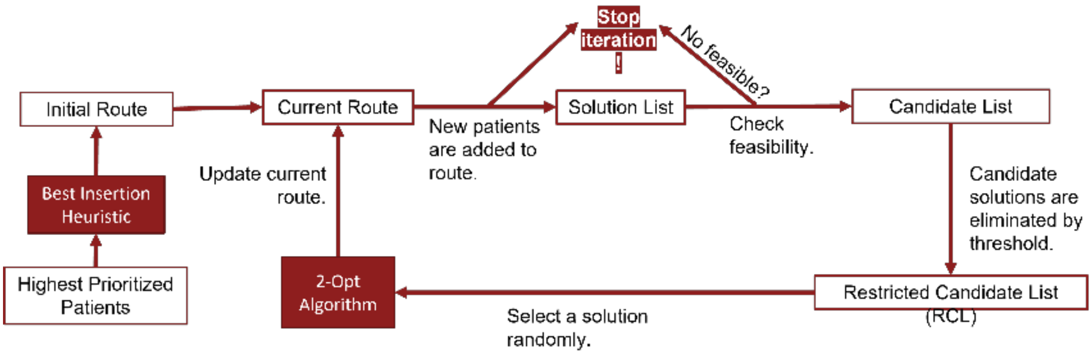


Figure 4.3: The flow chart of the GRASP heuristic algorithm.

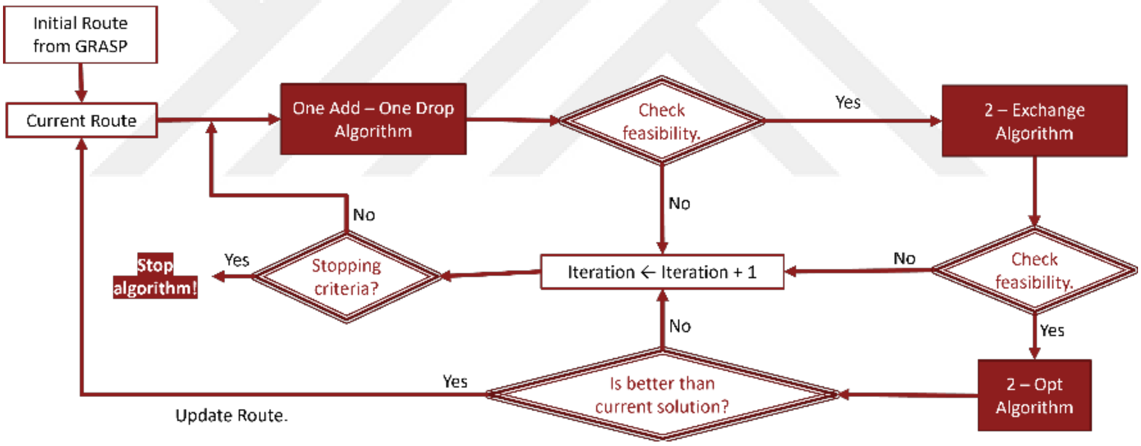


Figure 4.4: The flow chart of the VNS heuristic algorithm.

The GRASP+VNS Metaheuristic for the Static Problem

The inputs to the algorithm are each HSP's list of patients to be visited, service times, priority and location of the patients, travel time matrices calculated according to time blocks, and working hours of the personnel. Each of the parameters used in the heuristic algorithms has been determined after a detailed analysis and will be described in the computational study section.

The solution of the GRASP algorithm, used as a constructive heuristic, are input to the VNS algorithm with the purpose of improvement. Both heuristics are run in parallel in a multi-start fashion. Among the solutions obtained from parallel GRASP runs, the best one is fed to the VNS algorithm to be run in parallel and the best solution is kept.

GRASP Algorithm

At the beginning of the GRASP algorithm, the HSP has no patients or a route. Creating the route and the patient assignments is done step-by-step.

Firstly, the compulsory patients to be visited are assigned to the HSP using a greedy algorithm. Here, *the compulsory patients* refers to the highest-prioritized patients and the emergent patients. The greedy algorithm follows the best insertion method for reaching the minimum total time at the end of the algorithm. While performing insertion operations, the travel times and service times, which are time-dependent, are considered. Before running the greedy algorithm, the model checks whether infeasibility occurs due to total service times of the compulsory patients or not.

After scheduling the compulsory patients, the GRASP algorithm starts to assign the remaining patients to the HSP route greedily by considering the related objective. We designed a GRASP algorithm that works on loops. We called these loops as the "cycles" to avoid confusion with other terms. In each cycle, the GRASP constructive heuristic algorithm works by adding a patient until the algorithm can add no patients to the HSP's route, and is complete for a goal. It tries to add all the patients who are out of the route in each cycle, one by one, to the route which is at the beginning of the cycle and inserts the patient in the position that leads to the slightest increase in the total time of the route.

The local search for the insertion is performed by using the 2-opt algorithm. Then, a candidate list is created to represent the solutions with their scores. Score calculations in the candidate list are done separately for each objective. Solution qualities are determined by applying the scaling method according to the best objective function value of the solutions in the candidate list. The scaling method differs according to whether the sense of the objective function is maximization or minimization. If the objective function is maximized, the score of each solution is calculated as follows: $\frac{Obj.Value_x - Min.Obj.Value}{Max.Obj.Value - Min.Obj.Value}$. For minimization: $1 - \frac{Obj.Value_x - Min.Obj.Value}{Max.Obj.Value - Min.Obj.Value}$.

As we described, the solutions in the candidate list have scaled scores. The restricted candidate list (RCL) is created when the scaled scores of the solutions in the candidate list are eliminated with the determined threshold value. The threshold value is chosen between 0 and 1 as all scaled scores are normalized between 0 and 1. Then, a random solution is selected from the RCL, the cycle is terminated, and the next cycle is started. If no patients are left to run the new cycle, or if no patient cannot meet the feasibility check, the algorithm passes to the next objective. Since no solution provides a better objective function value for the previous goal, it is accepted as the best solution, and the objective function values are calculated for the other goals; otherwise, the best solution remains the same. All of these steps are detailed in the following pseudo-code.

Algorithm 1: Lexicographic Goal Programming - GRASP Heuristic - Single HSP Case

Input : D : Distance matrix, P : Patients, $P^{'}$: The highest prioritized patients, E : The beginning time of service providers, W : The ending time of service providers, T : The set of time blocks

Output: *Route* and *ArrivalTimes* arrays

```
Route ← {HPS};
ArrivalTimes ← {E};
for p ∈ P' do
    d ← +∞ ;
    startTime ← 0 ;
    for v ∈ Route do
        if v = HPS then
            time_v ← E ;
        else
            time_v ← p.arrivalTime + p.serviceTime ;
        endif
        for t ∈ T do
            if e_t ≤ time_v ≤ l_t then
                if d > D_tvp then
                    d ← D_tvp ;
                    startTime ← time_v ;
                endif
            endif
        end
    end
    p.arrivalTime ← d + startTime ;
    Route.append(p) ;
    ArrivalTime.append(p.arrivalTime) ;
end
v ← Route.length − 1;
startTime ← ArrivalTime[v] + Route[v].serviceTime ;
Route.append(HPS) ;
p ← Route.length − 1;
for t ∈ T do
    if e_t ≤ startTime ≤ l_t then
        d ← D_{t Route_v Route_p} ;
        ArrivalTime.append(startTime + d) ;
        break ;
    endif
end
bestPriority ← 0 ;
bestTotalTime ← +∞ ;
goal ← 0 ;
```

```
nonImprovingIter, Iter ← 1;
while goal ≠ 2 do
    tempRoute ← Route ;
    tempArrivalTime ← ArrivalTime ;
    while |P| ≠ |Route| do
        pos, rcl, rcl^l ← ∅;
        for p ∈ P do
            feasibilityCondition, candidateScore, candidateRoute,
              candidateArrivalTime ← graspFunc();
            if feasibilityCondition = True then
                pos.append(candidateScore);
                rcl.append((candidateRoute, candidateArrivalTime));
            end
        end
        if |pos| = 0 then
            break;
        end
        if goal = 0 then
            pos ← { posVal / max(pos) for posVal ∈ pos };
        else
            pos ← { 1 − (posVal − min(pos)) / (max(pos) − min(pos)) for posVal ∈ pos };
        end
        for posVal, candidateVal ∈ zip(pos, rcl) do
            if posVal ≥ 0.5 then
                rcl^l.append(candidateVal);
            end
        end
        selectedCandidate ← random(rcl^l);
        tempRoute, tempArrival ← 2-Opts(selectedCandidate);
    end
    goalPriority, goalTotalTime ← getObjectiveValues(tempRoute, tempArrivals) ;
    if (bestPriority < goalPriority) ∨ (bestPriority ≤ goalPriority ∧ bestTotalTime > goalTotalTime)
      then
        bestRoute ← tempRoute ;
        bestArrivalTime ← tempArrivalTime ;
        bestPriority ← goalPriority ;
        bestTotalTime ← goalTotalTime ;
        nonImprovingIter ← 0;
    end
    if Iter > iterationLimit^1 ∨ nonImprovingIter > iterationLimit^2 then
        goal ← goal + 1 ;
        nonImprovingIter, Iter ← 0;
    end
    nonImprovingIter ← nonImprovingIter + 1;
    Iter ← Iter + 1;
end
return   bestRoute,   bestArrivalTime,   bestPriority,   bestTotalTime
```

VNS Algorithm

The VNS algorithm consists of three main steps: shaking, local search, and changing the neighborhood. In our study, One add-One drop move and 2-Exchange move are performed for shaking, and the 2-Opt algorithm is implemented for local search. These moves will be explained the following paragraphs. Two variations of the VNS produced by alternating the order of the selected algorithms were tested and named VNS-1 and VNS-2. VNS-1 follows this cycle: 2-Exchange-One add-One drop-2 Opt, and VNS-2 follows this cycle: One add-One drop-2-Exchange-2 Opt. Since our tests showed that VNS-2 is a stronger heuristic, all narrations will be made through VNS-2. A detailed comparison of the two algorithms will be described in the computational study. The pseudo-code of the VNS-1 algorithm is available in the Appendix A.

The VNS algorithm takes the solution obtained from the GRASP algorithm as an initial solution. The VNS-2 algorithm starts with the One add-One drop move. Among the patients in the route, a random patient that does not have the highest priority is selected and removed from the initial route. Next, one of the patients postponed in the initial solution is chosen randomly and inserted in a random position in the route. In this way, the solutions obtained by changing the visited patients are given a chance. Thus, this move provides diversification for the first objective. The algorithm continues with the 2-Exchange move. Two random patients in the route are selected and their positions are exchanged. Shaking concludes with this step, and the algorithm passes to local search.

Local search implements 2-Opt moves. 2-Opt looks for a better solution by replacing all binary combinations that can be formed from patients on the route. If the algorithm finds a better solution, it updates the route and follows the same replacing procedure again. It continues until it cannot find a better solution. The improvement mentioned here is evaluated with respect to the objective functions in lexicographic goal programming.

After completing the local search, the obtained solution is compared to the current best solution. If it is not a better solution, then the iteration counters are increased by one, the best solution is not changed, and the cycle is rerun. If a better solution is found, the best solution is updated, the global counter is increased by one, the non-improving counter is reset, and the cycle is rerun. The loop continues until iteration counters exceed the defined stopping limits.

The pseudo-code of the algorithm is provided below.

Algorithm 2: Lexicographic Goal Programming - VNS Heuristic - Single HSP Case

Input : D : Distance matrix, P : Patients, P' : The highest-prioritized patients, E : The beginning time of service providers, W : The ending time of service providers, T : The set of time blocks, $Route$: The initial route from the GRASP heuristic, $ArrivalTimes$: The initial arrival times from the GRASP heuristic

Output: $Route$ and $ArrivalTimes$ arrays

$bestRoute \leftarrow Route$;
$bestArr \leftarrow ArrivalTimes$;
$iter$, $nonImprovingIter \leftarrow 0$;
while $iter < iteratonLimit^1 \wedge nonImprovingIter < iteratonLimit^2$ **do**
  $tempRoute \leftarrow bestRoute$;
  $tempArr \leftarrow bestArr$;
  $lastRoute$, $lastArr \leftarrow oneDrop(bestRoute, bestArr)$;
  $lastRoute$, $lastArr \leftarrow oneAdd(bestRoute, bestArr)$;
  $tempRoute \leftarrow bestRoute$;
  if $isBest(lastRoute, lastArr, bestRoute, bestArr)$ **then**
    $bestRoute \leftarrow lastRoute$;
    $bestArr \leftarrow lastArr$;
    $nonImprovingIter \leftarrow 0$;
  endif
  $lastRoute$, $lastArr \leftarrow twoExchange(lastRoute, lastArr)$;
  $lastRoute$, $lastArr \leftarrow twoOpts(lastRoute, lastArr, D, E, W)$;
  if $isBest(lastRoute, lastArr, bestRoute, bestArr)$ **then**
    $bestRoute \leftarrow lastRoute$;
    $bestArr \leftarrow lastArr$;
    $nonImprovingIter \leftarrow 0$;
  endif
  $nonImprovingIter \leftarrow nonImprovingIter + 1$;
  $iter \leftarrow iter + 1$;
end
$bestPriority$, $estTotalTime \leftarrow getObjectiveValues(bestRoute, bestArr)$;
return $bestRoute$, $bestArrivalTime$, $bestPriority$, $bestTotalTime$

The GRASP+VNS Metaheuristic Algorithm of the Dynamic Routing and Scheduling Problem with Multi-HSP Cooperation

The GRASP+VNS metaheuristic is similar to the GRASP+VNS metaheuristic algorithm described in section 4.2.1 except for several structural differences summarized as follows.

Recall that an initial route exists for each HSP. In addition to maximizing the total priority and minimizing the total time, there are also the objectives of minimizing the changes in the arrival times specified in the previous route and minimizing the overtime.

Two new constraints exist: not exceeding the overtime limit and visiting emergent patients within a deadline. The arrival time change value of the patients who were postponed is given as the maximum working hours per day to penalize postponement.

Another difference is that the algorithm simultaneously checks the routes of multiple HSPs. In the GRASP stage, there are more options for the patient to be added in each cycle. While performing diversification in VNS, a patient can be assigned to another HSP. This is performed in 2-Exchange, and One add-One drop moves.

GRASP Algorithm

The LGP GRASP algorithm tries to find the best solution for the lexicographically ordered goals. The algorithm solves the problem sequentially for goals ordered according to priorities. In each solution, the goal value of the previous solution is added to the model as a constraint. That is, the objective function values of the previous goals in the lexicographic order cannot get worse while the algorithm accepts the best solution for the current goal. We again note here that our lexicographically ordered objectives are to maximize total priority, minimize total arrival time changes, minimize total overtime, and minimize total time of the routes.

Before starting GRASP, the last locations and the first available times of the HSPs are collected from the last routes of the HSPs. These terms also were illustrated at Figure 3.1. The *"first available time"* refers to the time when the HSP has finished the current job and is ready to go to the new job. Then, this information is given to the GRASP algorithm as the starting node and the first available time for the first departure of HSPs' new routes. At time $t \in [9, 22]$, HSPs are either on a patient visit or on the road for a job. We called the time when the emergent call is arrived as "reOptTime". At the "reOptTime, the model first checks whether the HSPs are on their way or visiting. If an HSP is on the way at the "reOptTime", then "reOptTime" is considered to be his/her first available time. However, if an HSP is on a patient visit at that moment, him/her first available time is the completing time of the current visit since HSPs must complete the current visit. In any case, we assume HSPs were at the last patient location just before the "reOptTime".

Patients who should be visited are placed on the routes by a greedy algorithm. The greedy algorithm looks at the HSPs' routes to place the highest-prioritized or emergent patients. It aims to find the HSP route that, when the patient is added to the end of the HSP route, will cause the least increase in the total duration of all HSPs' routes. After the initial routes are created, the algorithm puts a non-highest-prioritized and non-emergent patient in one of the routes in each cycle of the algorithm, as described in the previous section. The patient to be placed is considered for all HSPs' route and insertion combinations. The candidate solutions have scaled scores to show their solution quality. The scaling process is the normalization of the goal values of the solutions so that the best value is 1 and the worst value is 0. The scaled goal values are named scaled scores. Then, the feasible candidate solutions are listed with their scaled scores in a candidate list. These candidates represent routes and schedules. It converts this candidate list to RCL by eliminating its scores according to the determined threshold. A candidate is arbitrarily selected from the obtained RCL. The cycle continues with the solution of the selected candidate. Therefore, the HSPs' routes and schedules are updated.

The feasibility of the solutions is checked according to whether the routes exceed the working hours plus the overtime limit, and whether the emergent calls are visited within the specified deadline. When calculating the score for RCL, for each goal, its own objective function value is taken into account.

If there is no patient to be added or if the stopping criteria, that is, the iteration limit and the non-improving iteration limit are exceeded, then the LGP GRASP algorithm stops. Thus, the result is returned since the search has been completed for all patients.

Below the pseudo-code of the GRASP algorithm is presented as Algorithm 3.

VNS Algorithm

Unlike the VNS described in the previous section, the routes of all HSPs are included in the search when diversifying. In the One add-One drop move, an arbitrarily selected patient of a random HSP is removed from the route, but there is no obligation to add patients to the same HSP when applying one add. Likewise, two patients swapped in the 2-Exchange procedure may be from different HSPs or the same HSP. However, the 2-Opt move is implemented to improve each HSP's route within itself.

In the algorithm, in each cycle, the obtained solution is compared with the best solution, where the objective functions "maximization of total priority", "minimization of total arrival time changes", "minimization of total overtime", and "minimization of total routes times" are used in lexicographic order.

Within a loop, One add-One drop, 2-Exchange, 2-Opt moves are implemented in each iteration in this order. As mentioned before, One add-One drop and 2-Exchange are used for diversification, and 2-Opt as a local search algorithm for intensification. The loop continues until one of the stopping conditions is satisfied, namely until the iteration limit or the non-improving iteration limit is exceeded. The pseudo-code is provided below.

4.3 Hybrid Solution Approach

As time progresses during the day, some patients have been already visited and thus, the size of the problem decreases. For this reason, we can try to utilize the MILP model, if it can provide an optimal solution within reasonable time. As a result, a hybrid solution approach is suggested here. As seen in Figure 4.5, if the MILP model can return the optimal solution in less than a minute, that solution is kept. However, if the solution it returns is not optimal, the GRASP+VNS metaheuristic algorithm is used to return a solution within seconds. The better one is selected as the valid solution. This is repeated for each dynamically incoming arrival. The algorithm stops at the end of the day (after the maximum allowed overtime is reached).

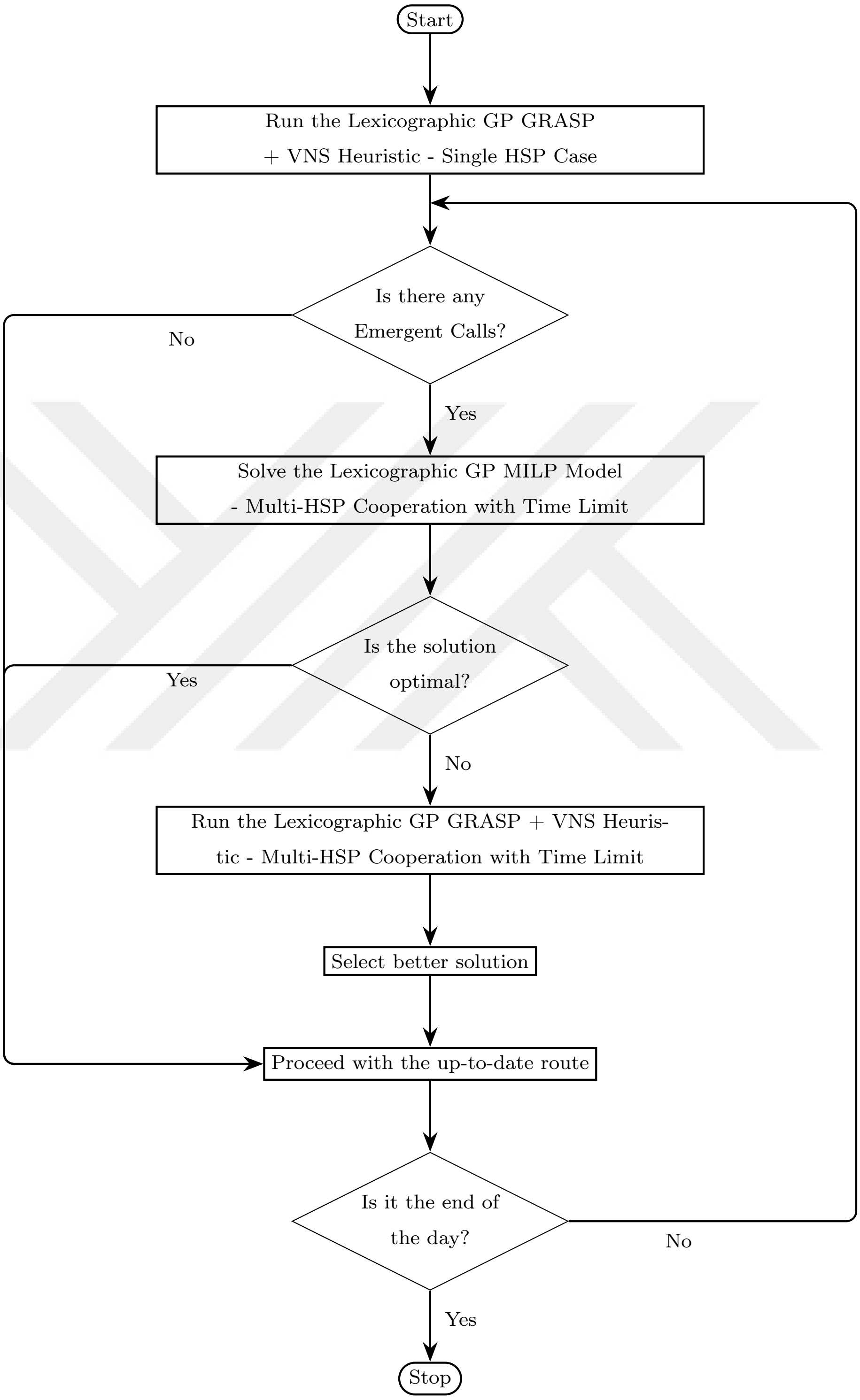


Figure 4.5: The hybrid solution approach procedure for a single day

Algorithm 3: Lexicographic Goal Programming - GRASP Heuristic - Multiple HSP Case

Input : D : Distance matrix, P : Patients, $P^{'}$: The high prioritized patients, P^{e} : The emergent patients, P^{c} : The current patients of the HSPs, N : The set of the HSPs, $ReOptTime$: The re-optimization time, W : The day ending time of service providers, U : The maximum allowed overtime of the HSPs, T : The set of time blocks

Output: *Routes* and *ArrivalTimes* arrays

```
for n ∈ N do
    Routes[n] ← {P^c[n]};
    ArrivalTimes[n] ← {P^c[n]};
end
for p ∈ P' do
    d ← +∞ ;                                                    ▷ P' ⊇ P^e
    previousTime ← 0 ;
    n ← None ;
    for n' ∈ N do
        previousTime' ← Routes[n'][−1].arrivalTime + Routes[n'][−1].serviceTime ;
        if previousTime' < ReOptTime then
            previousTime' ← ReOptTime;
        endif
        for t ∈ T do
            if e_t ≤ time_v ≤ l_t then
                if d > D_tvp then
                    d ← D_tvp ;
                    previousTime ← previousTime' ;
                    n ← n' ;
                endif
            endif
        end
    end
    p.arrivalTime ← d + previousTime ;
    Routes[n].append(p) ;
    ArrivalTimes[n].append(p.arrivalTime) ;
end
for n ∈ N do
    previousTime ← ArrivalTimes[n][−1] + Routes[n][−1] ;
    for t ∈ T do
        if e_t ≤ previousTime ≤ l_t then
            d ← D_t(Routes[n][−1])n ;
            ArrivalTimes[n].append(previousTime + d) ;
            break ;
        endif
    end
    Routes[n].append(n) ;
end
```

```
bestPriority, goal ← 0 ;
bestArrivalTimeChanges, bestOvertime, bestTotalTime ← +∞ ;
inRoute, bestInRoute ← P' ;
nonImprovingIter, Iter ← 1;
while goal ≠ 4 do
    tempInRoute ← inRoute ;
    tempRoutes ← Routes ;
    tempArrivalTimes ← ArrivalTimes ;
    while |P| ≠ |inRoute| do
        pos, rcl, rcl^l ← ∅;
        for p ∈ P inRoute do
            feasibilityCondition, candidateScore, candidateRoute,
              candidateArrivalTime ← graspFunc();
            if feasibilityCondition = True then
                pos.append(candidateScore);
                rcl.append((candidateRoute, candidateArrivalTime));
            end
        end
        if |pos| = 0 then
            break;
        end
        if goal = 0 then
            pos ← { posVal / max(pos) for posVal ∈ pos };
        else
            pos ← { 1 − (posVal − min(pos)) / (max(pos) − min(pos)) for posVal ∈ pos };
        end
        for posVal, candidateVal ∈ zip(pos, rcl) do
            if posVal ≥ 0.5 then
                rcl^l.append(candidateVal);
            end
        end
        selectedCandidate ← random(rcl^l);
        tempInRoute.append(selectedCandidate) ;
        tempRoute, tempArrival ← 2-Opts(selectedCandidate);
    end
    goalPriority, goalArrivalTimeChanges, goalOvertime,
      goalTotalTime ← getObjectiveValues(tempRoute, tempArrivals) ;
end
return bestRoute, bestArrivalTime, bestPriority, bestTotalTime
```

while ***cont'd*** **do**

  if $(bestPriority < goalPriority) \vee (bestPriority \leq goalPriority \wedge bestArrivalTimeChanges > goalArrivalTimeChanges) \vee (bestPriority \leq goalPriority \wedge bestArrivalTimeChanges \geq goalArrivalTimeChanges \wedge bestOvertime > goalOvertime) \vee (bestPriority \leq goalPriority \wedge bestArrivalTimeChanges \geq goalArrivalTimeChanges \wedge bestOvertime \geq goalOvertime \wedge bestTotalTime > goalTotalTime)$ **then**

    $bestRoute \leftarrow tempRoute$;

    $bestArrivalTime \leftarrow tempArrivalTime$;

    $bestPriority \leftarrow goalPriority$;

    $bestTotalTime \leftarrow goalTotalTime$;

    $nonImprovingIter \leftarrow 0$;

  end

  if $Iter > iterationLimit^1 \vee nonImprovingIter > iterationLimit^2$ **then**

    $goal \leftarrow goal + 1$;

    $nonImprovingIter, Iter \leftarrow 0$;

  end

  $nonImprovingIter \leftarrow nonImprovingIter + 1$;

  $Iter \leftarrow Iter + 1$;

end

Algorithm 4: Lexicographic Goal Programming - VNS Heuristic - Multiple HSPs Case

---

Input : D : Distance matrix, P : Patients, $P^{'}$: The highest-prioritized patients, $ReOptTime$: The re-optimization time, W : The ending time of service providers, U : The maximum allowed overtime of HSPs, T : The set of time blocks, $Routes$: The initial routes from GRASP heuristic, $ArrivalTimes$: The initial arrival times from the GRASP heuristic

Output: $Route$ and $ArrivalTimes$ arrays

$bestRoutes \leftarrow Routes$;
$bestArr \leftarrow ArrivalTimes$;
$iter$, $nonImprovingIter \leftarrow 0$;
while $iter < iteratonLimit^1 \wedge nonImprovingIter < iteratonLimit^2$ **do**
    $tempRoutes \leftarrow bestRoutes$;
    $tempArr \leftarrow bestArr$;
    $tempRoutes$, $tempArr \leftarrow oneDrop(tempRoutes, tempArr)$;
    $tempRoutes$, $tempArr \leftarrow oneAdd(tempRoutes, tempArr)$;
    if $isBest(tempRoutes, tempArr, bestRoutes, bestArr)$ **then**
        $bestRoute \leftarrow tempRoutes$;
        $bestArr \leftarrow tempArr$;
        $nonImprovingIter \leftarrow 0$;
    endif
    $tempRoutes$, $tempArr \leftarrow twoExchange(tempRoutes, tempArr)$;
    $tempRoutes$, $tempArr \leftarrow twoOpts(tempRoutes, tempArr, D, ReOptTime, W)$;
    if $isBest(tempRoutes, tempArr, bestRoutes, bestArr)$ **then**
        $bestRoutes \leftarrow tempRoutes$;
        $bestArr \leftarrow tempArr$;
        $nonImprovingIter \leftarrow 0$;
    endif
    $nonImprovingIter \leftarrow nonImprovingIter + 1$;
    $iter \leftarrow iter + 1$;
end
return $bestRoutes$, $bestArr$

Chapter 5

COMPUTATIONAL STUDY

In this chapter, we present a set of computational experiments on a COVID-19 case dataset with the solution approaches described in the previous section. After the COVID-19 virus has caused a global pandemic, countries' ministries of health and the World Health Organization (WHO) have listed a number of measures to prevent the spread of the COVID-19 virus. Treatment of people who are exposed to and infected with COVID-19 was also a major concern. World Health Organization (WHO) suggested home care for people with confirmed or suspected infection, if inpatient treatment was unavailable because of insufficient hospital capacity [World Health Organization,]. Home care was also advised for patients getting palliative care during the pandemic. During this pandemic, there were also patients whose treatments were carried out with home healthcare services. In highly populated cities, e.g., Istanbul, the inclusion of new urgent requests during the day in the daily plans by home healthcare services, emerged as a difficult problem to solve for service providers. Since this problem is similar to the Dynamic Home Healthcare Routing and Scheduling Problem (DHHCRSP), we used our data from the last DHHCRSP study [Cinar et al., 2022] by modifying it over a dataset for the COVID-19 case generated for a district in Istanbul.

The first section in this chapter explains the process for generating the dataset for emergent patients, generation of Istanbul's traffic data for our region under consideration and the data structures used in the computational study. In the following sections, the parameter settings in GRASP and VNS algorithms that we use for solving both the dynamic and static problems are described. Two different VNS algorithms are designed, which are called as VNS-1 and VNS-2. In this study, VNS algorithms' performance comparison is also performed. In addition, the results of the MILP and GRASP+VNS approaches are also compared for the static problem.

Computer Properties	
RAM	128 GB
Processor	Intel(R) Xeon(R) W-2295
CPU	3.00GHz
# Core	18
# Core (Logical)	36

Software/IDE	Version
Python	3.8.8
PyCharm (IDE)	2021.2.3
IBM CPLEX ILOG	20.1.0

Table 5.1: Information on the computing platform used in the computational study.

Then, we compare our developed approaches for the dynamic problem, as in the static problem. MILP & MILP, MILP & GRASP+VNS, GRASP+VNS & MILP, GRASP+VNS & GRASP+VNS and GRASP+VNS & HYBRID solution approaches for the dynamic problem are analyzed. In the last section, the effect of HSPs cooperation, which we state as our main contribution in this study, on the solution quality is examined in detail.

In this chapter, the GAP values in the tables show the values obtained from CPLEX for MILP models. For the GRASP+VNS algorithm and the Hybrid approach, it shows the difference with MILP solutions. Since both static and dynamic problems are lexicographically ordered multiobjective problems, GAP values are calculated according to this order. For MILP models, the GAP values of the goal with the first GAP value are shown. For the heuristic approaches, the GAP value for the first goal, which is different from the MILP model, is shown.

All computational runs were performed using the computer for which the specifications are presented in Table 5.1. For MILP, CPLEX solver was used in parallel with 28 cores, with a time limit of an hour. GRASP+VNS solution algorithm was coded in Python and executed on the same computer. GRASP and VNS algorithms

Static Problem			
Goal	**Definition**	**Max/Min**	**Unit**
Goal 1	Total Priority	Maximization	
Goal 2	Total Time	Minimization	Hours

Dynamic Problem			
Goal	**Definition**	**Max/Min**	**Unit**
Goal 1	Total Priority	Maximization	
Goal 2	Arrival Time Changes	Minimization	Hours
Goal 3	Total Overtime	Minimization	Hours
Goal 4	Total Time	Minimization	Hours

Table 5.2: Summary of the goals used in the static and dynamic problems.

were also run in parallel with 28 cores.

5.1 Data Generation

Our data sets were based on several data sets created for a DHHCRSP study conducted by Cinar et al., [Cinar et al., 2022]. The data sets in [Cinar et al., 2022] were created using clustering and prediction models according to the number of COVID-19 patients announced by the Republic of Turkey, Ministry of Health. One district of Istanbul (Kağıthane) was selected and partitioned such that patient clusters were formed according to geographic proximity. Then, individual patient locations were created in these clusters randomly according to COVID-19 positive case densities. Finally, each HSP was assigned to a cluster. The data includes patient locations, estimated treatment times and patient priorities. The data set consists of nine HSPs and 1156 patients. In this study, since we were doing a computational study for two HSPs, we selected two neighboring HSPs' clusters. These two HSPs were in charge of a total of 240 patients. There were 119 patients for HSP1 and 121 patients for HSP2.

For each HSP, 30 samples were generated for each combination of patient sizes (20, 40, 60), and dynamism levels (Low, Mid, High). In total, 3x3x30=270 samples were created for an HSP. Since we used two HSPs, we obtained 540 samples. We added emergent patients to the processed data. Emergent patients were created by choosing among the patients who could not be taken to the route during the day after solving the static problem. Arrival times of selected emergent patients were randomly generated with a uniform distribution between 10:00 and 17:00. In addition, the priority values of emergent patients were set to 5, which was the highest value. Before modifying the data, each patient has a priority between 1 and 5. Recall that in our dynamic problem priority values have two indices, namely patient index and HSP index. In order to create second HSP index, we assigned a second priority value to the HSP to which the patient was not initially assigned. If the initial priority value was 5 which was the highest value, the second priority value was also 5. However, if not, the second priority value was one less than the initial priority value. After this assignment process, we had 2-dimensional priority data. 2-dimensions was done to ensure care continuity. Patient location data is shown in Figure 5.1.

We set our model inputs as follows. We solved the problem sets described above for a single day with two HSPs. The working hours of HSPs are between 09:00 and 18:00. In addition, they have a maximum of 4-hour overtime only for the dynamic problem. Data sets are created in order to see how the approaches would perform against dynamism in the dynamic problem and patient density in the static problem. For the performance analysis of the static problem, we created patient sets of 20, 40, and 60 patients for each HSP. These patients were randomly selected from HSPs' own patient clusters for them respectively. Data sets for the dynamic problem were created with two main groups; the initial routes that were obtained from the solutions to the static problem, and emergent patients. The number of emergent patients varies according to the dynamism level, respectively, 4 for low-level, 6 for mid-level, 8 for high-level. Different emergent patient numbers were designed to test the model performance under different dynamism conditions. The emergent patients' call times were created using a uniform distribution between

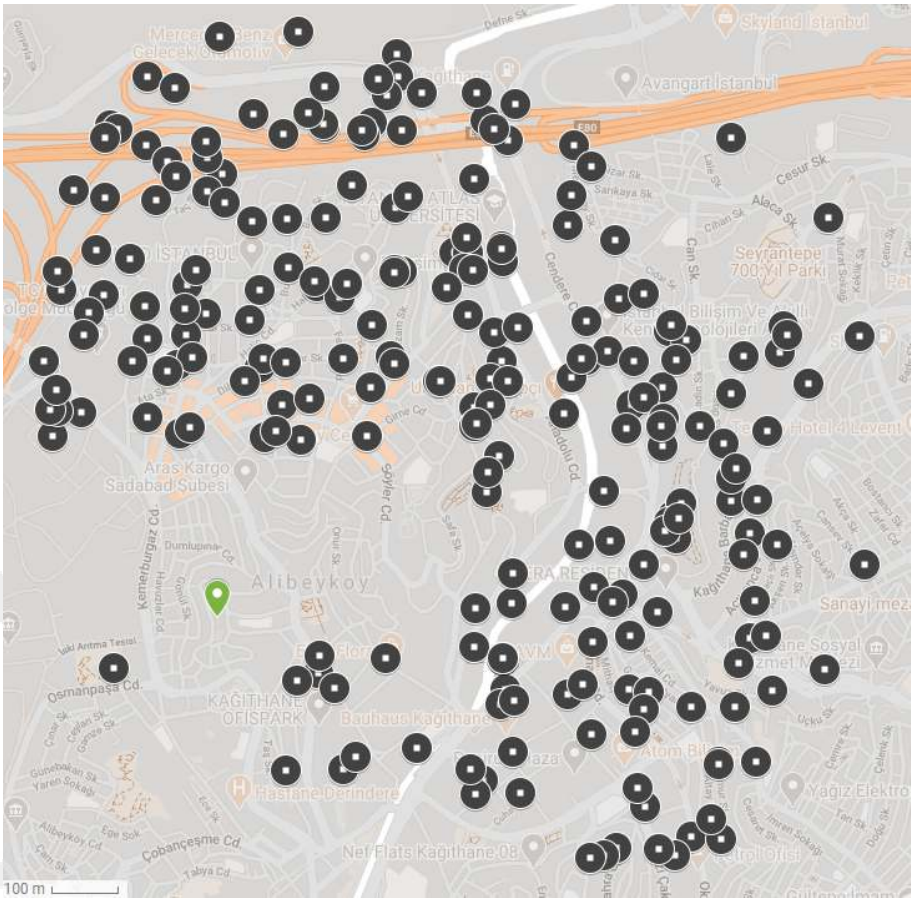


Figure 5.1: The patients locations on the map.

10:00 and 17:00. As explained in the previous chapter, the model was re-run with each emergent arrival call.

5.1.1 Istanbul Traffic Data

In order to make our problem setting more realistic, we generated the travel duration between patients in the data set according to the traffic situation depending on time of the day. We used a 3-dimensional duration matrix in both static and dynamic problems. The third dimension represents the time intervals (we refer to them as blocks in our model description). We divided the daily work hours into three separate intervals. We used the travel times in these intervals, both in our mathematical model and heuristics, according to the beginning time of the intervals, that is, 09:00, 13:00 and 17:00. We pulled the traffic data from Bing.com via an API at these time points to calculate the travel times.

5.2 Parameter Settings

The parameter hyper-tuning was required to improve the performance of the heuristics. Grid-search method, which is a valid method for hyper tuning, is preferred. In the Grid search method, certain values are given for different parameters. The values that give the best results among all possible combinations of these values are used as parameter values. Analyzes of grid search studies for both GRASP and VNS heuristics are shown.

At the same time, if there are alternatives to each other in the algorithms used, a choice should be made between them. As we mentioned in Chapter 4, we have both VNS-1 and VNS-2 heuristics. Since we prefer VNS-2 as a result of the analysis, all VNS mentioned in the following sections refer to the VNS-2 heuristic.

5.3 The Selection of VNS Heuristics

In order to choose which one to use between VNS-1 and VNS-2, 30 samples, which have size 20 and low-level dynamism, were run. In these runs, the static problem was solved. Goal 1 and Goal 2 represent total priority and total time, respectively, as shown in Table 5.2. The results obtained from the solutions are presented in Table 5.3.

Since there is a time limit of an hour when solving MILP in Table 5.3, if the time limit reaches the Goal 1 in LGP, it does not return a GAP value for Goal 2. These non-occurring GAP values were indicated by "-". The "Elapsed Time" column is in seconds. The "Best Solution" column shows which of the MILP, GRASP+VNS-1 and GRASP+VNS-2 approaches returned the better solution. The "Best VNS" column shows which was better between GRASP+VNS-1 and GRASP+VNS-2. If a solution is a better solution, it is first checked whether the Goal 1 value is greater than LGP. If Goal 1 values are equal, the lower Goal 2 value is selected. If both Goals are equal, the solution with the smaller Elapsed Time is said to be better.

As can be seen from the "Best VNS" column, VNS-1 returns better results at 13 samples and VNS-2 at 18 samples. Although they are close to each other, we can say that VNS-2 performs better. Therefore, VNS-2 is preferred.

Examining the Best Solution column, it can be seen that there are 13 samples where GRASP+VNS outperforms MILPs. In cases where they cannot find a better solution, the average GAP is 1.56% for both GRASP+VNS types. Thus, the insight is obtained that the selected GRASP+VNSs have a solution quality close to MILPs in the static problem.

Table 5.3: The Static Problem Results: Comparison for MILP, GRASP+VNS-1 and GRASP+VNS-2.

Simulation No	MILP			GRASP + VNS 1			GRASP + VNS 2			Best Solution	Best VNS
	GAP		Elapsed	GAP		Elapsed	GAP		Elapsed		
	Goal 1	Goal 2	Time	Goal 1	Goal 2	Time	Goal 1	Goal 2	Time		
0	3.00%	-	3601.30	0.00%	-2.46%	471.24	0.00%	-2.57%	461.18	VNS-2	VNS-2
1	4.01%	-	3612.75	2.50%	-3.12%	478.27	2.50%	-3.40%	486.70	MILP	VNS-2
2	3.94%	-	3606.59	0.00%	-1.62%	476.28	0.00%	-1.67%	475.19	VNS-2	VNS-2
3	0.00%	1.65%	43.08	0.00%	0.06%	411.28	0.00%	0.01%	409.43	MILP	VNS-2
4	0.00%	1.68%	1.77	2.44%	-0.97%	536.45	2.44%	-0.75%	525.50	MILP	VNS-1
5	0.00%	1.98%	19.55	0.00%	-0.12%	483.87	0.00%	-0.23%	485.84	VNS-2	VNS-2
6	0.00%	1.96%	52.78	2.33%	-0.31%	532.00	2.33%	-0.42%	533.44	MILP	VNS-2
7	0.00%	1.34%	1.98	0.00%	0.06%	472.88	0.00%	0.00%	470.49	MILP	VNS-2
8	0.00%	1.33%	3.84	0.00%	0.44%	461.37	0.00%	0.44%	461.37	MILP	VNS-2
9	3.94%	-	3620.39	0.00%	-2.16%	460.72	0.00%	-1.99%	456.94	VNS-1	VNS-1
10	4.89%	-	3609.97	0.00%	-0.94%	498.18	0.00%	-0.88%	489.52	VNS-1	VNS-1
11	0.00%	1.95%	726.39	2.27%	-0.14%	473.77	2.27%	-0.30%	467.94	MILP	VNS-2
12	0.00%	2.00%	12.66	2.38%	-1.43%	481.61	2.38%	-1.37%	485.81	MILP	VNS-1
13	2.85%	-	3601.25	0.00%	0.59%	500.89	0.00%	0.31%	499.99	MILP	VNS-2
14	0.00%	1.62%	198.70	2.56%	-1.95%	495.54	2.56%	-1.38%	485.17	MILP	VNS-1
15	0.00%	1.82%	18.59	0.00%	-0.37%	489.29	0.00%	-0.43%	485.82	VNS-2	VNS-2
16	0.00%	1.61%	8.81	0.00%	0.42%	472.69	0.00%	0.48%	473.21	MILP	VNS-1
17	0.00%	2.08%	29.70	0.00%	-0.55%	482.99	0.00%	-0.61%	484.23	VNS-2	VNS-2
18	0.00%	1.11%	14.97	0.00%	-0.21%	477.54	0.00%	-0.21%	470.59	VNS-2	VNS-2
19	0.00%	1.93%	48.52	0.00%	-0.27%	431.80	0.00%	-0.11%	428.67	VNS-1	VNS-1
20	0.00%	2.05%	2628.33	2.44%	-0.89%	471.45	2.44%	-1.00%	474.19	MILP	VNS-2
21	0.00%	1.16%	12.05	0.00%	-0.10%	466.22	0.00%	0.02%	462.25	VNS-1	VNS-1
22	0.00%	1.38%	30.09	0.00%	-0.20%	473.28	2.22%	-0.15%	473.50	VNS-1	VNS-1
23	3.90%	-	3610.36	-2.44%	1.04%	470.18	-2.44%	1.04%	473.98	VNS-1	VNS-1
24	0.00%	1.57%	19.58	2.38%	-1.52%	512.26	2.38%	-1.63%	513.79	MILP	VNS-2
25	3.59%	-	3605.84	2.50%	-1.98%	501.71	2.50%	-2.10%	507.49	MILP	VNS-2
26	0.00%	1.00%	3.41	2.38%	-0.77%	430.49	2.38%	-0.71%	434.55	MILP	VNS-1
27	0.00%	0.93%	3.45	2.33%	-0.20%	537.82	2.33%	-0.20%	540.54	MILP	VNS-1
28	0.00%	1.77%	37.94	0.00%	0.73%	482.37	0.00%	0.62%	481.21	MILP	VNS-2
29	0.00%	1.52%	1.45	0.00%	-0.05%	507.83	0.00%	-0.28%	505.75	VNS-2	VNS-2

5.4 GRASP + VNS Metaheuristic for the Static Problem

As we mentioned in the introduction of this section, it is a method used for grid search parameters hypertuning and trying all combinations. We get the results shown in Table 5.4 by running four parameters that we use in GRASP+VNS with five samples each. The continuation of the results is presented in Table 5.5. The values shown here are the averages of the RCL Ratio and Iteration Limit values on the left and the values taken for five samples when the model is run. Unlike the "Max Run Time" shown in the last column, it shows the highest run time value in five samples. In GRASP, as stopping criteria, a restriction is given to both non-improving iterations and a restriction to the total number of iterations. In VNS, there is a stopping criteria only for non-improving iterations. Run Time values are in seconds. The samples used are respectively "40-Low-HSP1-0", "40-Low-HSP1-1", "40-Low-HSP1-2", "40-Low-HSP1-3", "40-Low-HSP1- 4".

The results are Goal 1, Goal 2, Average Run Time is given in order according to Maximum Run Time. In Table 5.4, the best Goal values are obtained with the combination of *0.5-50-10-200*. However, since both the average and maximum run time values are above the threshold of 60 seconds, they are not considered as parameter values. Instead, the combination of *0.5-50-50-100*, which reaches the second best goal value approximately four times faster, is chosen. Since the total iteration number cannot be exceeded, the "Non Improving" iteration limit for GRASP is removed in the combination of the *0.5-50-50-100* parameter.

0.5-50-50-100 parameter values will be used in all solutions for the static problem.

Table 5.4: The Grid Search Results of the GRASP + VNS Parameters for the Static Problem.

RCL Ratio	Iteration Limit			Goal 1	Goal 2	Avg Run Time	Max Run Time
	GRASP		VNS				
	Stopping	NonImproving	NonImproving				
0.5	50	10	200	45.40	17.75	140.23	427.11
0.5	**50**	**50**	**100**	**45.40**	**17.81**	**38.59**	**45.22**
0.5	100	50	200	45.40	17.81	101.37	176.29
0.5	200	50	50	45.40	17.82	54.40	62.43
0.3	50	50	100	45.40	17.82	36.59	46.70
0.7	50	50	100	45.40	17.83	36.98	45.58
0.5	200	50	100	45.40	17.85	58.74	65.85
0.5	100	50	100	45.20	17.67	60.16	77.02
0.3	50	30	100	45.20	17.72	36.97	56.73
0.3	200	30	200	45.20	17.74	106.29	178.42
0.5	100	50	50	45.20	17.76	48.90	59.08
0.7	200	50	200	45.20	17.76	108.91	178.54
0.7	200	10	200	45.20	17.77	102.33	187.03
0.7	200	30	100	45.20	17.78	39.48	49.23
0.7	200	30	50	45.20	17.79	35.48	41.55
0.7	200	30	200	45.20	17.80	75.45	130.07
0.7	50	50	200	45.20	17.81	76.56	125.24
0.7	100	30	100	45.20	17.81	40.82	49.16
0.7	100	10	50	45.20	17.81	14.60	17.80
0.7	50	50	50	45.20	17.81	32.63	37.65
0.7	200	10	100	45.20	17.82	20.44	32.10
0.7	200	50	100	45.20	17.82	60.53	67.33
0.7	50	30	50	45.20	17.83	28.48	31.82
0.7	100	30	50	45.20	17.84	34.16	39.54
0.7	50	30	100	45.00	17.67	41.28	54.82
0.3	200	50	200	45.00	17.67	140.14	219.83
0.3	100	50	100	45.00	17.67	58.89	79.24
0.5	50	50	200	45.00	17.67	99.32	166.36
0.5	200	50	200	45.00	17.67	107.01	166.85
0.7	100	50	50	45.00	17.68	50.12	60.63
0.3	50	30	200	45.00	17.68	87.30	162.94
0.3	50	50	200	45.00	17.68	129.50	221.15
0.5	100	30	100	45.00	17.68	44.45	62.08
0.3	200	30	50	45.00	17.68	35.66	41.73
0.3	50	50	50	45.00	17.68	34.22	40.58
0.7	100	50	100	45.00	17.68	62.07	84.34
0.7	100	50	200	45.00	17.68	120.27	203.52
0.3	200	50	100	45.00	17.69	65.34	86.59
0.5	200	30	200	45.00	17.69	97.42	153.74
0.5	50	10	50	45.00	17.71	15.85	20.41
0.5	50	30	200	45.00	17.71	93.19	182.33

5.5 *GRASP + VNS Metaheuristic for the Dynamic Problem*

We use the grid search method for the Dynamic problem, like the GRASP+VNS hypertuning which we did for the static problem. "40-High-0" sample has been

Table 5.5: The Grid Search Results of the GRASP + VNS Parameters for the Static Problem.- *Cont'd*

RCL Ratio	Iteration Limit			Goal 1	Goal 2	Avg Run Time	Max Run Time
	GRASP		VNS				
	Stopping	NonImproving	NonImproving				
0.7	200	50	50	45.00	17.71	57.09	67.51
0.3	100	50	50	45.00	17.71	48.95	59.01
0.3	100	30	100	45.00	17.71	44.81	63.43
0.3	200	50	50	45.00	17.72	56.04	62.15
0.7	100	10	200	45.00	17.72	99.71	184.28
0.5	100	30	200	45.00	17.72	107.63	174.43
0.3	100	50	200	45.00	17.72	120.74	185.08
0.7	50	10	100	45.00	17.73	21.68	31.80
0.3	100	30	50	45.00	17.73	33.78	42.33
0.5	200	10	100	45.00	17.73	23.95	40.37
0.7	100	10	100	45.00	17.74	22.75	33.70
0.5	200	30	50	45.00	17.74	36.46	41.80
0.5	50	50	50	45.00	17.74	33.33	41.73
0.7	200	10	50	45.00	17.77	17.33	25.28
0.7	100	30	200	45.00	17.79	71.42	128.78
0.7	50	10	200	45.00	17.79	62.54	113.52
0.7	50	30	200	45.00	17.79	113.78	209.11
0.3	200	10	50	45.00	17.80	14.79	16.59
0.3	50	30	50	45.00	17.82	28.31	31.06
0.5	50	10	100	44.80	17.64	28.64	45.87
0.7	50	10	50	44.80	17.64	17.91	24.73
0.3	50	10	200	44.80	17.67	67.33	105.80
0.5	100	30	50	44.80	17.68	35.24	42.29
0.3	200	30	100	44.80	17.69	40.70	57.87
0.3	100	30	200	44.80	17.70	91.75	234.95
0.5	100	10	100	44.80	17.70	24.93	36.24
0.5	50	30	50	44.80	17.71	29.43	35.29
0.5	200	10	50	44.60	17.59	17.63	20.49
0.3	100	10	200	44.60	17.60	79.58	147.10
0.3	200	10	100	44.60	17.61	25.98	35.01
0.3	100	10	50	44.60	17.61	16.71	20.00
0.5	50	30	100	44.60	17.61	40.56	52.46
0.3	50	10	100	44.60	17.62	27.50	41.02
0.5	100	10	50	44.60	17.62	17.55	21.43
0.5	200	30	100	44.60	17.63	46.94	58.87
0.3	100	10	100	44.60	17.69	23.81	34.90
0.5	200	10	200	44.40	17.59	73.35	129.70
0.5	100	10	200	44.40	17.67	72.11	114.10
0.3	200	10	200	44.20	17.56	97.16	140.25
0.3	50	10	50	44.20	17.57	17.29	23.74

solved. The initial route is used from the solutions obtained by MILP. Since the dynamism level is high, solutions have been prepared for eight emergent patients. Eight re-optimizations are performed for eight emergent patients. A "Performance

Score" is obtained from the results of eight re-optimizations. The performance score is obtained by multiplying the rankings of the goals with the coefficient values given according to the priorities of the goals. The calculation is done as follows.

$$\text{Score} = \#\text{ Goal 1 } * 8 + \#\text{ Goal 2 } * 4 + \#\text{ Goal 3 } * 2 + \#\text{ Goal 4 } * 1 \tag{5.1}$$

Combinations with any re-optimization run time higher than 60 seconds are eliminated before performance scores are evaluated. The remaining values are given in Table 5.6. Among these values, the combination of parameters *0.7-100-10-100*, which has the lowest score, is selected.

At Table 5.6, it is seen that the NonImproving value of GRASP is lower than the value of Static. This can be interpreted as more improvements have been made with VNS in the problems solved on the Dynamic side. In this way, it can be deduced that even if GRASP as a construction algorithm gives a bad result, it can be improved with VNS.

0.7-100-10-100 parameters values are used in all dynamic GRASP+VNS meta-heuristics mentioned in the following sections.

5.6 Results for the Static Problem

In the Static Problem, individual initial routes are calculated for each HSP. The difference in the data used is the sizes. For each HSP, a single day visit plan was drawn from the patient clusters of 20, 40, and 60 diseases. CPLEX run with 1 hour time limit for solving the Static Problem. GRASP+VNS metaheuristic algorithm were coded on Python, and also .

In the Static Problem, HSPs do not have the right to overtime. Also, their Goals are also different from the Dynamic Problem. Its goals are total priority maximization and total time minimization as shown in Table 5.2.

The performance of MILP and GRASP+VNS in terms of Goal 1 is shown in Figure 5.2. The values here show which method is more successful in how many samples. "EQUAL" represents that both methods reach the same value for Goal 1.

Table 5.6: The Grid Search Elected Results of the GRASP + VNS Parameters for the Dynamic Problem.

RCL Ratio	Iteration Limit			Performance Score
	GRASP		VNS	
	Stopping	NonImproving	NonImproving	
$*$ 0.7	**100**	**10**	**100**	**799**
0.3	100	10	100	993
0.5	100	10	100	994
0.3	200	10	50	1027
0.3	50	10	50	1148
0.7	200	10	30	1149
0.5	50	10	100	1149
0.3	50	10	30	1168
0.7	100	10	50	1204
0.3	200	10	100	1210
0.5	100	10	50	1222
0.5	100	10	30	1239
0.3	100	10	50	1260
0.7	50	10	100	1262
0.5	200	10	100	1280
0.5	200	10	30	1298
0.3	50	10	100	1300
0.3	100	10	30	1318
0.7	50	10	30	1319
0.3	200	10	30	1345
0.7	200	10	50	1379
0.5	200	10	50	1435
0.5	50	10	30	1487
0.5	50	10	50	1583

As the sample size increases, the rate of GRASP+VNS finding the result obtained in MILP for Goal 1 decreases. However, even in the lowest HSP2-60 and HSP2-40 samples, it is observed that half of the samples run reach the result obtained in MILP. In total, GRASP+VNS achieves better than MILP in 3% of the solutions it brought and the same solution as MILP in 38%. In addition, in the results where MILP is better, our average GAP values are as follows; HSP1-20: 3.98%, HSP2-20: 4.13%, HSP1-40: 4.71%, HSP2-40: 4.17%, HSP1-60: 4.35%, HSP2-60: 3.30%.

An inference can be made from the section, using the GRASP+VNS metaheuris-

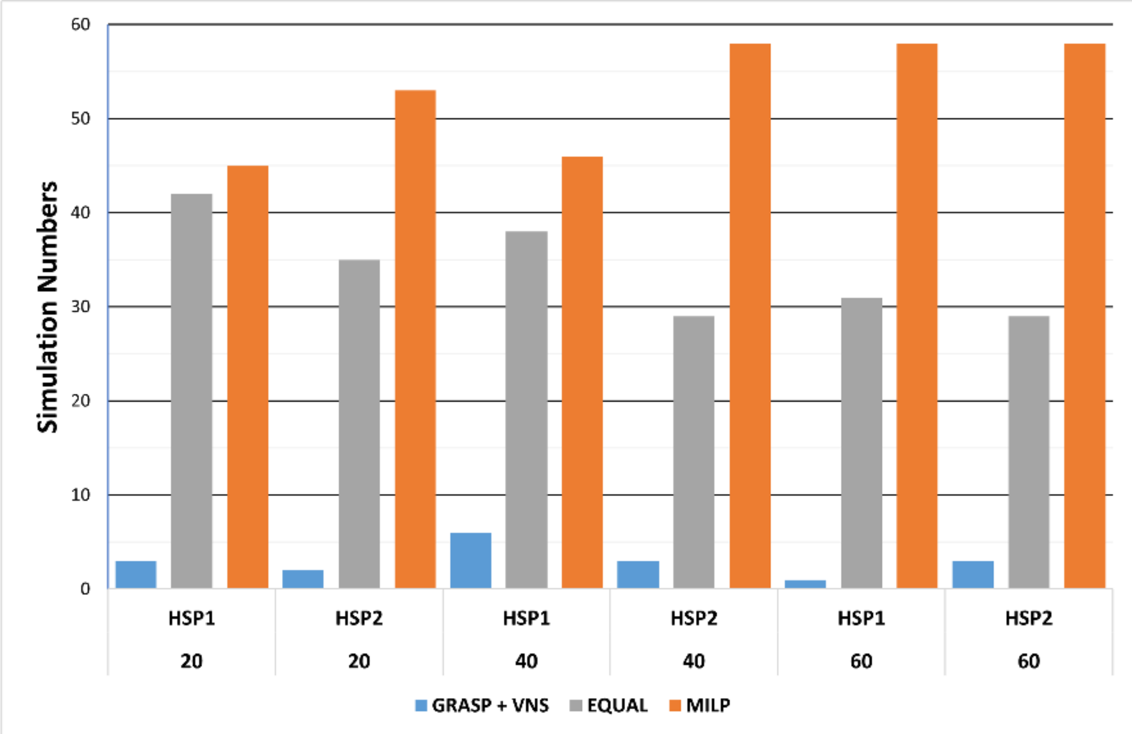


Figure 5.2: The comparison of the better solution counts of the MILP and GRASP+VNS methods according to size values and HSPs.

tic algorithm, with an average GAP value of 4.01% in Total Priority, a better Total Time value can be achieved with 18.4 times smaller run time on the average. At the same time, a similar number of patients are served in these solutions.

The solutions obtained in the static problem are given in Table 5.7, Table 5.8, and Table 5.9. In these tables, the "Size" column shows how many patients the models made for a day's planning. The "Dynamism Level" column represents the emergent patient numbers of the relevant datasets in the dynamic part.The emergent patient numbers are two patients per HSP for "Low", three patients per HSP for "Mid", and four patients per HSP for "High". We add this column for use when examining the results in the dynamics section. The "Dynamism Level" column is of no importance in this section, as it does not contribute to the static problem. For this reason, Table 5.10 analysis, which we grouped over sizes, is made. The "HSP" column represents which HSP the datasets belong to. Run time values are in seconds. "# Post." column shows the postponed patients. The GAP values represent the difference

Table 5.7: Simulation results: Static Problem Comparisons

Size	Dynamism Level	HSP	Stats.	MILP					GRASP + VNS				
				Run Time	**Goal 1**	**Goal 2**	**# Post.**	**GAP**	**Run Time**	**Goal 1**	**Goal 2**	**# Post.**	**GAP**
20	**Low**	**HSP1**	Avg	394.5	42.8	17.9	9.6	4.3%	24.6	42.6	17.8	9.5	1.5%
			Min	1.0	40.0	17.7	9.0	4.1%	21.4	39.0	17.5	9.0	0.0%
			Max	3610.6	47.0	18.0	10.0	4.6%	27.7	47.0	18.0	10.0	2.5%
			Std Dev	1091.0	1.9	0.1	0.5	0.2%	2.1	1.9	0.2	0.5	1.0%
		HSP2	Avg	1092.9	41.5	17.9	9.7	3.8%	25.5	41.2	17.7	9.6	1.8%
			Min	1.5	36.0	17.6	9.0	2.9%	19.7	36.0	17.3	9.0	0.0%
			Max	3620.4	45.0	18.0	10.0	4.9%	30.7	45.0	18.0	10.0	2.6%
			Std Dev	1617.0	2.5	0.1	0.5	0.6%	2.8	2.5	0.2	0.5	1.0%
	Mid	**HSP1**	Avg	448.8	43.4	17.9	9.5	4.2%	24.9	42.8	17.8	9.4	4.3%
			Min	0.8	38.0	17.6	9.0	2.7%	20.1	40.0	17.4	9.0	0.3%
			Max	3628.5	47.0	18.0	10.0	5.9%	35.3	48.0	18.0	10.0	12.8%
			Std Dev	1139.1	2.0	0.1	0.5	1.6%	3.3	1.6	0.2	0.5	3.9%
		HSP2	Avg	878.5	41.9	17.9	9.6	4.4%	26.2	40.4	17.7	9.5	5.0%
			Min	4.2	37.0	17.5	9.0	3.2%	21.0	37.0	17.3	9.0	0.1%
			Max	3645.1	46.0	18.0	10.0	6.8%	30.0	44.0	18.0	10.0	17.8%
			Std Dev	1483.5	2.5	0.1	0.5	1.3%	2.6	1.7	0.2	0.5	5.1%
	High	**HSP1**	Avg	208.2	43.9	17.9	9.4	4.3%	24.4	42.5	17.8	9.4	5.3%
			Min	0.7	41.0	17.6	9.0	4.3%	18.8	38.0	17.3	9.0	0.1%
			Max	3606.6	48.0	18.0	10.0	4.3%	29.7	45.0	18.0	10.0	14.6%
			Std Dev	721.8	2.0	0.1	0.5	-	2.6	1.6	0.2	0.5	4.7%
		HSP2	Avg	656.5	41.9	17.9	9.5	4.1%	25.9	40.6	17.7	9.6	4.7%
			Min	2.7	38.0	17.7	9.0	3.1%	20.2	37.0	17.3	9.0	0.1%
			Max	3616.5	45.0	18.0	10.0	5.8%	30.5	44.0	18.0	10.0	13.6%
			Std Dev	1232.8	1.6	0.1	0.5	1.3%	2.9	1.8	0.2	0.5	4.2%

between the objective function values of Goal 1. If the objective function is the same in Goal 1, it represents the difference in Goal 2. If both are the same, the GAP value returns zero and these zero values are not included in the statistical calculations. The GAP values given for MILP are the values provided by CPLEX

Table 5.8: Simulation results: Static Problem Comparisons.-*Cont'd*

Size	Dynamism Level	HSP	Stats.	MILP Run Time	MILP Goal 1	MILP Goal 2	MILP # Post.	MILP GAP	GRASP + VNS Run Time	GRASP + VNS Goal 1	GRASP + VNS Goal 2	GRASP + VNS # Post.	GRASP + VNS GAP
40	**Low**	**HSP1**	Avg	468.1	48.7	17.9	29.0	5.2%	35.7	48.1	17.7	29.2	3.3%
			Min	9.0	45.0	17.8	28.0	4.9%	14.7	45.0	17.3	29.0	2.0%
			Max	3622.1	52.0	18.0	30.0	5.5%	52.7	50.0	18.0	30.0	6.0%
			Std Dev	1061.4	1.6	0.1	0.3	0.4%	8.0	1.4	0.2	0.4	1.5%
		HSP2	Avg	951.9	45.7	17.9	29.2	6.0%	39.8	44.8	17.7	29.3	2.6%
			Min	10.4	41.0	17.7	29.0	3.4%	26.7	40.0	17.2	29.0	0.3%
			Max	3768.5	50.0	18.0	30.0	8.4%	48.1	49.0	18.0	30.0	4.7%
			Std Dev	1404.5	1.8	0.1	0.4	1.8%	6.1	2.0	0.2	0.5	1.2%
	Mid	**HSP1**	Avg	707.9	48.2	17.9	29.1	5.3%	37.9	47.3	17.8	29.1	5.2%
			Min	6.6	43.0	17.6	29.0	3.1%	7.0	45.0	17.4	29.0	0.3%
			Max	3638.2	52.0	18.0	30.0	7.8%	51.6	49.0	18.0	30.0	11.8%
			Std Dev	1350.9	2.2	0.1	0.3	1.8%	9.3	1.2	0.2	0.3	3.1%
		HSP2	Avg	873.8	45.9	18.0	29.1	6.9%	39.6	44.9	17.8	29.1	5.5%
			Min	21.9	44.0	17.8	29.0	6.5%	30.3	41.0	17.3	29.0	0.0%
			Max	3607.2	51.0	18.0	30.0	7.3%	50.2	48.0	18.0	30.0	15.7%
			Std Dev	1174.9	1.7	0.0	0.3	0.5%	4.9	1.5	0.2	0.3	4.8%
	High	**HSP1**	Avg	678.3	48.2	17.9	29.0	5.0%	35.3	46.8	17.7	29.3	5.1%
			Min	1.9	43.0	17.7	29.0	2.4%	14.8	43.0	17.3	29.0	0.0%
			Max	3684.3	52.0	18.0	30.0	8.1%	49.5	51.0	18.0	30.0	15.4%
			Std Dev	1345.9	2.0	0.1	0.2	2.2%	8.2	1.8	0.2	0.5	4.6%
		HSP2	Avg	619.9	46.3	18.0	29.0	2.5%	40.2	44.6	17.8	29.1	4.9%
			Min	17.1	43.0	17.8	29.0	2.5%	28.9	42.0	17.3	29.0	0.0%
			Max	3601.1	49.0	18.0	29.0	2.5%	52.8	49.0	18.0	30.0	10.6%
			Std Dev	948.3	1.5	0.0	0.0	-	5.3	1.6	0.2	0.3	3.6%

and since there is a an-hour time limit, GAP values are returned. The GAP value provided for GRASP+VNS shows the values compared to MILP.

When the MILP model is run as LGP, if it obtains the Goal 1 value as optimal, it usually has obtained an optimal solution for all given models, since it restricts the

Table 5.9: Simulation results: Static Problem Comparisons.-*Cont'd*

Size	Dynamism Level	HSP	Stats.	MILP					GRASP + VNS				
				Run Time	Goal 1	Goal 2	# Post.	GAP	Run Time	Goal 1	Goal 2	# Post.	GAP
60	Low	HSP1	Avg	572.1	51.2	17.9	48.9	3.9%	38.2	49.7	17.6	49.3	4.0%
			Min	11.0	46.0	17.6	48.0	2.2%	9.7	46.0	17.3	49.0	1.9%
			Max	3606.1	54.0	18.0	49.0	5.6%	71.8	53.0	18.0	50.0	5.9%
			Std Dev	1213.3	1.7	0.1	0.3	1.5%	15.7	1.5	0.2	0.5	1.7%
		HSP2	Avg	806.3	48.3	18.0	49.1	6.4%	46.9	47.6	17.8	49.2	2.6%
			Min	7.2	44.0	17.7	49.0	3.2%	21.3	44.0	17.3	49.0	2.0%
			Max	3869.4	51.0	18.0	50.0	9.3%	61.2	50.0	17.9	50.0	4.3%
			Std Dev	1201.9	1.6	0.1	0.3	2.5%	8.4	1.6	0.2	0.4	0.9%
	Mid	HSP1	Avg	397.3	51.2	17.9	49.0	4.0%	35.1	49.6	17.7	49.3	4.2%
			Min	14.0	48.0	17.7	48.0	3.8%	9.6	48.0	17.3	49.0	1.9%
			Max	3609.3	53.0	18.0	49.0	4.2%	63.8	53.0	18.0	50.0	7.7%
			Std Dev	916.1	1.2	0.1	0.2	0.3%	15.3	1.2	0.2	0.5	1.9%
		HSP2	Avg	562.5	48.2	18.0	49.0	6.1%	48.6	47.3	17.8	49.1	3.5%
			Min	4.6	45.0	17.9	49.0	6.1%	29.4	45.0	17.4	49.0	0.0%
			Max	3608.2	51.0	18.0	50.0	6.1%	68.1	50.0	18.0	50.0	9.8%
			Std Dev	1004.2	1.6	0.0	0.2	-	9.4	1.1	0.2	0.3	2.5%
	High	HSP1	Avg	411.8	51.1	18.0	48.9	4.1%	32.1	49.9	17.7	49.4	4.9%
			Min	28.3	48.0	17.9	48.0	3.6%	8.6	47.0	17.4	49.0	2.0%
			Max	3658.3	53.0	18.0	49.0	4.6%	59.3	52.0	18.0	50.0	11.3%
			Std Dev	889.3	1.4	0.0	0.3	0.7%	14.2	1.3	0.2	0.5	2.4%
		HSP2	Avg	864.3	48.2	18.0	49.0	5.4%	48.0	47.0	17.8	49.1	3.9%
			Min	5.2	46.0	17.7	49.0	4.1%	20.6	44.0	17.4	49.0	0.0%
			Max	3615.0	51.0	18.0	50.0	7.9%	62.1	50.0	18.0	50.0	8.0%
			Std Dev	1213.1	1.2	0.1	0.2	1.7%	8.6	1.6	0.2	0.3	2.3%

solution set for Goal 2. Thus, the GAP values shown for MILP only cover Goal 1.

When we subtract the number of postponed patients from the size value, we get

Table 5.10: Simulation results: Static Problem Comparisons by Sample Sizes

Size	HSP	Stats.	MILP					GRASP + VNS				
			Run Time	Goal 1	Goal 2	# Post. Patients	GAP	Run Time	Goal 1	Goal 2	# Post. Patients	GAP
20	HSP1	Avg	350.50	43.39	17.92	9.51	4.27%	24.65	42.66	17.79	9.44	3.98%
		Min	0.72	38.00	17.62	9.00	2.69%	18.83	38.00	17.30	9.00	0.04%
		Max	3628.50	48.00	18.00	10.00	5.90%	35.29	48.00	17.99	10.00	14.58%
		Std Dev	995.53	2.00	0.10	0.50	0.94%	2.70	1.71	0.18	0.50	4.01%
	HSP2	Avg	875.98	41.74	17.88	9.60	4.04%	25.84	40.72	17.70	9.56	4.13%
		Min	1.45	36.00	17.45	9.00	2.85%	19.70	36.00	17.25	9.00	0.02%
		Max	3645.09	46.00	18.00	10.00	6.77%	30.72	45.00	18.00	10.00	17.78%
		Std Dev	1447.90	2.21	0.12	0.49	1.03%	2.73	2.04	0.23	0.50	4.27%
40	HSP1	Avg	618.11	48.38	17.93	29.06	5.18%	36.30	47.40	17.73	29.19	4.71%
		Min	1.94	43.00	17.59	28.00	2.38%	6.97	43.00	17.29	29.00	0.03%
		Max	3684.30	52.00	18.00	30.00	8.14%	52.72	51.00	18.00	30.00	15.38%
		Std Dev	1250.42	1.96	0.09	0.27	1.74%	8.51	1.56	0.19	0.39	3.57%
	HSP2	Avg	815.21	45.98	17.95	29.09	5.82%	39.90	44.76	17.78	29.17	4.17%
		Min	10.42	41.00	17.68	29.00	2.46%	26.70	40.00	17.21	29.00	0.01%
		Max	3768.47	51.00	18.00	30.00	8.37%	52.79	49.00	18.00	30.00	15.69%
		Std Dev	1185.71	1.70	0.06	0.29	1.94%	5.38	1.72	0.19	0.37	3.45%
60	HSP1	Avg	460.39	51.19	17.96	48.92	3.96%	35.01	49.72	17.70	49.34	4.35%
		Min	11.03	46.00	17.59	48.00	2.16%	8.59	46.00	17.31	49.00	1.92%
		Max	3658.30	54.00	18.00	49.00	5.59%	71.85	53.00	18.00	50.00	11.32%
		Std Dev	1008.53	1.42	0.06	0.27	1.06%	15.08	1.32	0.21	0.48	1.99%
	HSP2	Avg	744.35	48.27	17.96	49.06	5.91%	47.86	47.30	17.80	49.11	3.30%
		Min	4.59	44.00	17.69	49.00	3.17%	20.63	44.00	17.34	49.00	0.01%
		Max	3869.38	51.00	18.00	50.00	9.28%	68.12	50.00	18.00	50.00	9.80%
		Std Dev	1138.48	1.47	0.06	0.23	1.92%	8.72	1.47	0.16	0.32	2.03%

the number of patients served. Thus, for all size values, it is seen that a maximum of 12 patients and a minimum of 10 patients are planned at the beginning of the day. Considering that the service times of the patients are generated with $\mathcal{N}(45, 2)$, and the HSPs works between 9.00-18.00 without overtime, it is expected that the HSPs visits about these served patients numbers.

Figure 5.3 shows the average goal values obtained by the HSPs with MILP and GRASP+VNS approaches for the static problem for different sample sizes. Total Priority reflects Goal 1 and Total Time reflects Goal 2. It is clear from the figure that MILP has achieved more successful results for Total Priority. On the other hand, GRASP+VNS seems to have been more successful in the Total Time. Since

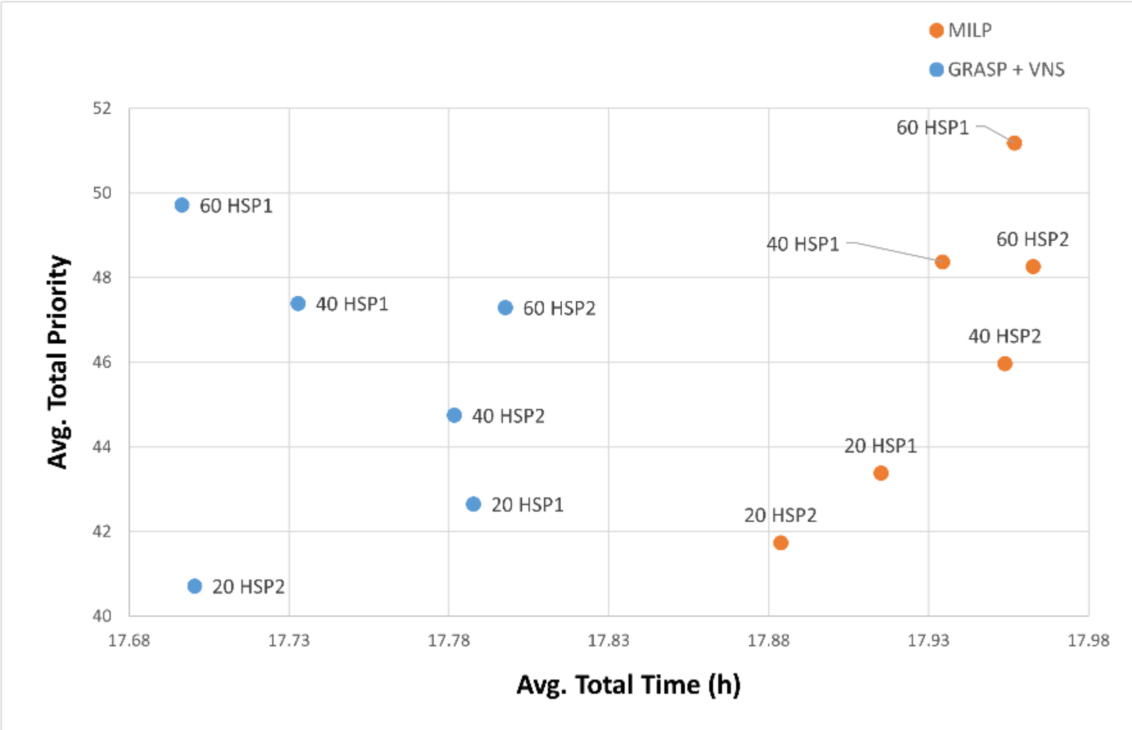


Figure 5.3: The average values of the total priority and the total time obtained from size and HSP samples according to MILP and GRASP+VNS methods in the static problem.

LGP is applied, success in Goal 1, i.e. Total Priority, is more significant.

5.7 *Results of the Dynamic Problem*

The initial routes obtained in the static initial stage of the Dynamic Problem are merged, and we try to include the dynamically occurring emergent patients in the route with the cooperation of the HSPs. While trying to solve this problem, the model has goals the following: maximizing total priority, minimizing arrival time changes, minimizing overtime, and minimizing total time. The LGP model is solved in the given order.

As we mentioned in Chapter 4, we have MILP, GRASP+VNS and HYBRID solution methods for the dynamic problem. For the static problem, as we mentioned in the previous section, we had MILP and GRASP+VNS solutions. In order to make

Static Problem	**Dynamic Problem**	**Combined**
MILP	MILP	MILP & MILP
MILP	GRASP+VNS	MILP & GRASP+VNS
GRASP+VNS	MILP	GRASP+VNS & MILP
GRASP+VNS	GRASP+VNS	GRASP+VNS & GRASP+VNS
GRASP+VNS	HYBRID	GRASP+VNS & HYBRID

Table 5.11: Summary of which methods were used for static and dynamic problem solving in combined methods.

the results of the dynamic part and the comparisons we present more detailed, we have created the combinations specified in Table 5.11. In this section, these combinations are analyzed. Our goal is to choose the most efficient approach.

The datasets used as inputs consist of the initial plans of the HSPs and then the emergent patients created according to the dynamism levels. Dynamism levels are calculated over 20 patients at the beginning of the HSP. Since we are making a solution with a dataset containing two HSPs, the number of emergent patients in the problem is four, six, eight, according to their low, mid, high dynamism levels, respectively. As we stated in the Data Generation section, emergent patient request times are established for these patients between 10:00 and 17:00. With every emergent patient request, the model re-optimizes the problem.

While evaluating the results shown in this section, the results obtained are from the routes and schedules at the end of the day, making it difficult to analyze. Even if the solution of each re-optimization MILP model was optimal, it can cause to move away from the optimal solution for the whole day because of the robust structure of the dynamic problem. To reach the optimal solution for the whole day, we should know when the emergent cases will occur, which is impossible. We try to approximate the optimal solution with our proposed five methods.

In the dynamic problem, the overtime rights of HSPs are determined as four hours. There are four goals used in LGP used in Dynamic Problem. As shown in Table 5.2, the lexicographically ordered goals are respectively maximization of total priority, minimization of arrival time changes, minimization of overtime and minimization of total time.

In the dynamic problem, the MILP model was run in parallel with 28 cores, with a time limit of one minute in CPLEX. The GRASP+VNS metaheuristic algorithm was coded in Python with 28 cores and run in parallel with a 1-minute time limit. Both GRASP and VNS were run in parallel.

Average goal values obtained from 30 samples in size and dynamism level values given in Table 5.12 are shown. The "Size" column contains 20, 40, 60. These values refer to the size of the sample used when creating the initial plan in the static problem. When the Goal 2 column representing the total time in Table 5.10 is examined, it is observed that a tighter route is formed as the size increases. The tight route means the route which has the total route time close to day-end time. The new patients cannot be added to these routes without overtime. We add this column to our analysis in order to analyze the effect of starting the day with a tight route according to the solutions we propose for the dynamic problem. The "Dynamism Level" column refers to the number of emergent patients and thus the number of re-optimizations. As mentioned at the beginning of this section, there are three levels Low, Mid, and High. These dynamism levels are corresponding to four, six, and eight emergent patients per two HSPs, respectively. The values in the goal column are given in Table 5.2.

When comparing the presented solution approaches for Goal 1, MILP & MILP is found to be the best solution in most size and dynamism level values. However, GRASP+VNS & GRASP+VNS and MILP & GRASP+VNS solutions also follow with a difference of 5% for Goal 1. In fact, there are samples where GRASP+VNS & GRASP+VNS and MILP & GRASP+VNS are more successful than MILP & MILP, 20-High, 20-Low, respectively.

Goal 2 results are directly affected by Goal 1 results since LGP is applied. In fact, when a model is run, a selection is made among the alternate optimal solutions of the first priority goal value. The LGP approach aims to find the best solution for Goal 2 among the alternate optimal solutions of the Goal 1. Here again, MILP & MILP method seems to be the most successful one. We will mention it as a weakness

Table 5.12: Simulation results: Dynamic Problem Comparisons.

Size	Dynamism Level	Goals	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID
20	Low	Goal 1	104.30	104.45	103.37	103.87	100.20
		Goal 2	20.84	27.63	27.39	22.01	27.56
		Goal 3	3.60	2.47	2.25	3.55	1.87
		Goal 4	39.60	38.27	37.85	39.55	37.87
	Mid	Goal 1	115.27	112.83	110.93	115.00	109.04
		Goal 2	32.29	38.11	36.04	32.98	39.68
		Goal 3	5.56	4.35	3.21	5.51	3.88
		Goal 4	41.56	40.06	38.82	41.51	39.88
	High	Goal 1	123.90	121.73	112.31	125.04	116.48
		Goal 2	17.45	56.55	50.12	50.96	58.46
		Goal 3	2.76	5.67	3.90	7.19	5.34
		Goal 4	38.76	41.55	39.17	43.19	41.34
40	Low	Goal 1	114.37	114.43	112.24	112.97	110.07
		Goal 2	23.00	32.55	28.49	24.05	28.66
		Goal 3	3.80	2.82	2.21	3.68	1.98
		Goal 4	39.80	38.65	38.04	39.68	37.98
	Mid	Goal 1	124.13	123.77	118.89	122.86	117.57
		Goal 2	37.56	45.82	42.20	38.87	45.41
		Goal 3	5.67	4.52	3.21	5.62	3.91
		Goal 4	41.67	40.52	38.84	41.62	39.91
	High	Goal 1	133.92	128.76	128.15	132.85	123.17
		Goal 2	53.27	58.67	53.88	54.50	61.51
		Goal 3	7.62	5.55	4.77	7.42	5.56
		Goal 4	43.62	41.40	40.64	43.42	41.56
60	Low	Goal 1	119.50	119.32	114.54	117.33	113.80
		Goal 2	20.24	28.37	25.71	21.28	27.04
		Goal 3	3.79	2.86	2.14	3.71	2.13
		Goal 4	39.79	38.70	37.50	39.71	38.13
	Mid	Goal 1	129.50	128.47	123.17	127.72	122.19
		Goal 2	37.52	45.18	42.88	38.21	44.72
		Goal 3	5.75	4.47	3.48	5.62	3.95
		Goal 4	41.75	40.44	39.02	41.62	39.95
	High	Goal 1	138.96	135.40	132.12	137.12	127.46
		Goal 2	52.45	56.03	53.42	52.59	60.35
		Goal 3	7.58	5.46	4.43	7.47	5.68
		Goal 4	43.58	41.46	40.14	43.47	41.68

in Chapter 6 as one of the reasons why MILP & MILP method is so successful. GRASP+VNS algorithm sorts things one after another, but the MILP model can shift routes if it satisfies end-of-day and emergent patient deadline constraints. In this way, arrival time change values can be reduced even more.Except for the 20-High example, the method has a maximum of 5% slight difference between MILP & MILP and GRASP+VNS & GRASP+VNS. In the samples where the MILP & MILP solution is better than GRASP+VNS & GRASP+VNS solution, there is an average 1.87% GAP between them for Goal 1.

There is a clearly visible correlation between Goal 3 and Goal 4. The main reason for this, of course, is that the model is forced to overtime with newly arriving emergent patients. Since the allowed overtime value is used in Goal 1 and Goal 2, solutions usually have overtime value. In Table 5.12 and Table 5.13, there are rows where *Goal 4* is not equal to 36 + *Goal 3*. The reason is that the total time of an HSP does not reach 36, but the other performs overtime. In these examples, it can be said that one HSP has been assigned more patients than the other with newly arrived emergent patients, or it can be said that he/she has worked longer due to the time on the road.

Contrary to other goal values in Goal 3 and Goal 4 solutions, the best results are achieved by the GRASP+VNS & MILP and GRASP+VNS & HYBRID methods. Failures in Goal 1 and Goal 2, that is, fewer patients are served, resulting in a shorter total route time and less overtime.

A detailed analysis of the goal values of the methods applied in the dynamic problem can be seen from Table 5.13. To summarize, in general, the standard deviation values of the methods are close to each other except GRASP+VNS & HYBRID. Since the standard deviation value of the GRASP+VNS & HYBRID, it is not as consistent as other models.

In Figure 5.4, for goal and run time values among all samples, the best result is represented by 1, and the worst result is represented by 0. In this way, the average values are normalized using the minimum and maximum values. The Run Time value taken here represents the sum of all re-optimization run time values for a sample until the end of the day. Data in Figure 5.4 is obtained by averaging over all samples.

Run Time values will be examined in detail in the following sections. However,

Table 5.13: Simulation results: Dynamic Problem Detailed Comparisons.

Size	Dyna. Level	Stats.	Goal 1					Goal 2					Goal 3					Goal 4				
			MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID	MILP &	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID
20	Low	Avg	104.30	104.45	103.37	103.87	100.20	20.84	27.63	27.39	22.01	27.56	3.60	2.47	2.25	3.55	1.87	39.60	38.27	37.85	39.55	37.87
		Min	98.00	98.00	92.00	98.00	92.00	11.34	14.85	11.11	11.65	16.92	2.82	1.61	0.77	2.83	0.98	38.82	37.56	35.93	38.83	36.98
		Max	110.00	110.00	109.00	109.00	109.00	30.34	42.22	43.73	29.55	38.50	4.43	3.30	3.25	4.27	3.50	40.43	38.83	39.25	40.27	39.50
		Std Dev	3.06	3.00	3.93	3.17	5.70	4.20	6.01	6.97	4.38	5.00	0.39	0.44	0.59	0.42	0.55	0.39	0.35	0.65	0.42	0.55
	Mid	Avg	115.27	112.83	110.93	115.00	109.04	32.29	38.11	36.04	32.98	39.68	5.56	4.35	3.21	5.51	3.88	41.56	40.06	38.82	41.51	39.88
		Min	110.00	53.00	78.00	110.00	96.00	22.40	24.92	19.89	22.07	24.73	4.94	0.00	0.00	4.74	2.91	40.94	27.53	31.55	40.74	38.91
		Max	122.00	122.00	120.00	121.00	120.00	47.45	50.99	45.99	47.85	60.46	6.37	6.15	4.47	6.63	5.48	42.37	42.15	40.47	42.63	41.48
		Std Dev	3.05	11.70	8.94	3.01	5.79	5.42	7.06	5.89	5.98	7.74	0.38	0.98	1.17	0.56	0.74	0.38	2.43	2.01	0.56	0.74
	High	Avg	123.90	121.73	112.31	125.04	116.48	17.45	56.55	50.12	50.96	58.46	2.76	5.67	3.90	7.19	5.34	38.76	41.55	39.17	43.19	41.34
		Min	105.00	87.00	57.00	121.00	97.00	7.43	44.38	35.20	36.78	46.15	0.70	0.00	0.00	6.41	4.19	36.70	32.20	32.34	42.41	40.19
		Max	133.00	130.00	131.00	132.00	127.00	49.58	77.97	63.03	64.44	74.92	7.01	6.66	6.08	7.91	6.36	43.01	42.66	42.08	43.91	42.36
		Std Dev	6.04	7.56	19.29	2.82	6.09	10.92	7.79	6.35	6.52	6.51	1.24	1.31	2.33	0.40	0.44	1.24	1.90	3.44	0.40	0.44
40	Low	Avg	114.37	114.43	112.24	112.97	110.07	23.00	32.55	28.49	24.05	28.66	3.80	2.82	2.21	3.68	1.98	39.80	38.65	38.04	39.68	37.98
		Min	111.00	107.00	104.00	107.00	101.00	12.84	22.62	15.41	13.66	20.07	3.26	0.93	1.30	2.55	0.74	39.26	36.93	37.07	38.55	36.74
		Max	120.00	120.00	117.00	117.00	117.00	33.68	48.08	48.78	32.72	36.77	4.37	3.43	3.17	4.20	3.39	40.37	39.43	39.02	40.20	39.39
		Std Dev	2.41	2.66	2.97	2.50	4.51	5.34	5.94	6.97	5.46	4.76	0.31	0.49	0.51	0.39	0.49	0.31	0.54	0.43	0.39	0.49
	Mid	Avg	124.13	123.77	118.89	122.86	117.57	37.56	45.82	42.20	38.87	45.41	5.67	4.52	3.21	5.62	3.91	41.67	40.52	38.84	41.62	39.91
		Min	119.00	118.00	94.00	118.00	105.00	22.76	28.49	20.51	24.27	28.30	5.18	3.31	0.43	4.73	2.69	41.18	39.31	34.78	40.73	38.69
		Max	132.00	131.00	127.00	129.00	125.00	48.81	55.66	61.35	51.13	58.86	6.23	5.32	4.80	6.48	4.68	42.23	41.32	40.80	42.48	40.68
		Std Dev	2.84	2.96	6.81	2.90	4.63	5.82	6.23	9.84	6.52	7.54	0.30	0.46	1.11	0.41	0.56	0.30	0.46	1.46	0.41	0.56
	High	Avg	133.92	128.76	128.15	132.85	123.17	53.27	58.67	53.88	54.50	61.51	7.62	5.55	4.77	7.42	5.56	43.62	41.40	40.64	43.42	41.56
		Min	129.00	94.00	106.00	128.00	99.00	40.43	34.07	35.46	40.86	49.90	6.94	0.00	0.38	5.93	4.25	42.94	32.27	35.23	41.93	40.25
		Max	139.00	136.00	137.00	138.00	136.00	62.43	72.08	71.26	71.16	78.49	8.00	6.55	6.55	8.01	6.59	44.00	42.55	42.55	44.01	42.59
		Std Dev	2.42	9.18	7.31	2.38	6.57	6.49	8.30	9.39	7.18	7.59	0.33	1.66	1.32	0.48	0.63	0.33	2.24	1.59	0.48	0.63
60	Low	Avg	119.50	119.32	114.54	117.33	113.80	20.24	28.37	25.71	21.28	27.04	3.79	2.86	2.14	3.71	2.13	39.79	38.70	37.50	39.71	38.13
		Min	114.00	114.00	99.00	113.00	107.00	12.34	15.75	12.08	13.10	18.27	3.30	1.87	0.00	2.88	1.05	39.30	37.67	33.85	38.88	37.05
		Max	125.00	125.00	121.00	121.00	121.00	29.42	43.92	43.33	32.48	63.21	4.36	3.75	3.31	4.48	6.40	40.36	39.75	38.56	40.48	42.40
		Std Dev	2.20	2.24	5.13	1.84	4.63	4.48	6.92	8.08	5.71	9.05	0.27	0.36	0.63	0.30	0.98	0.27	0.48	1.17	0.30	0.98
	Mid	Avg	129.50	128.47	123.17	127.72	122.19	37.52	45.18	42.88	38.21	44.72	5.75	4.47	3.48	5.62	3.95	41.75	40.44	39.02	41.62	39.95
		Min	126.00	123.00	89.00	124.00	109.00	26.24	25.48	22.75	24.07	27.13	5.18	1.24	1.22	5.05	2.83	41.18	37.24	32.40	41.05	38.83
		Max	133.00	133.00	131.00	132.00	132.00	48.13	59.77	67.27	51.23	56.16	6.38	5.84	4.67	6.28	5.67	42.38	41.84	40.67	42.28	41.67
		Std Dev	1.64	2.31	8.50	2.32	5.78	5.72	9.13	10.36	6.78	6.71	0.27	0.91	0.89	0.31	0.63	0.27	0.97	1.72	0.31	0.63
	High	Avg	138.96	135.40	132.12	137.12	127.46	52.45	56.03	53.42	52.59	60.35	7.58	5.46	4.43	7.47	5.68	43.58	41.46	40.14	43.47	41.68
		Min	135.00	119.00	110.00	133.00	117.00	41.67	33.03	29.29	34.80	34.80	6.96	1.38	2.37	6.42	4.65	42.96	37.38	36.60	42.42	40.65
		Max	143.00	142.00	139.00	141.00	137.00	68.73	75.71	77.93	73.25	80.54	8.00	6.47	6.55	8.00	7.48	44.00	42.47	42.55	44.00	43.48
		Std Dev	1.86	5.15	6.36	1.87	6.08	7.66	9.45	10.72	8.62	9.69	0.28	1.26	1.15	0.47	0.69	0.28	1.26	1.53	0.47	0.69

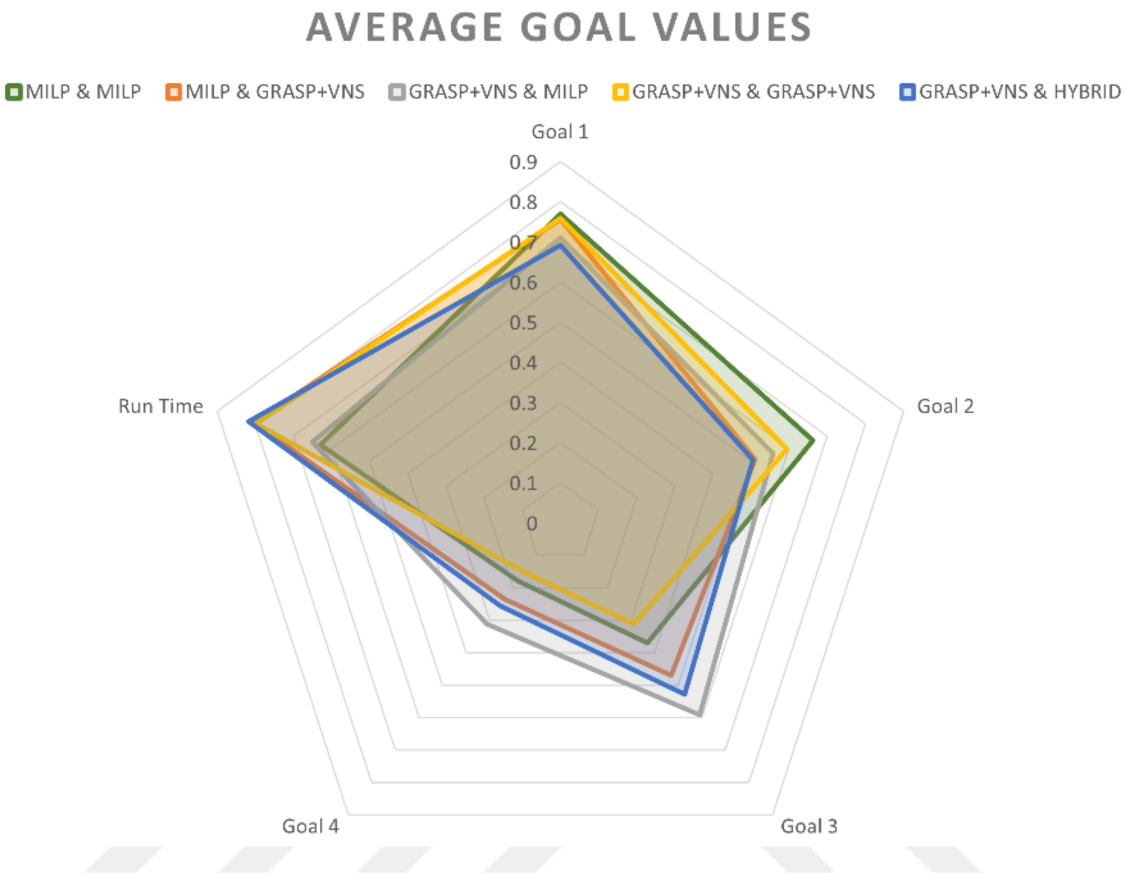


Figure 5.4: The Comparison of Methods by Goal 1, Goal 2, Goal 3, Goal 4 and Run Time.

it will be useful to examine the run time while discussing the methods in Figure 5.4. As we have seen here and mentioned in the results of Table 5.12, there does not seem to be much difference between MILP & MILP, GRASP+VNS & MILP and GRASP+VNS & GRASP+VNS methods in terms of goal values. When examined together with the run time values, the performance of the GRASP+VNS & GRASP+VNS method stands out.

Table 5.14, which results in a higher priority value as more patients were visited initially, as expected in the sample size results. However, as an effect of this, arrival time changes, overtime and total time values aimed at minimization have also increased. The table in which the statistical analysis of these values is compared is available in Appendix B, Table B.1.

Spider charts of Figure 5.4 broken down into sizes are given in Figure 5.5. As

Table 5.14: Simulation results: Dynamic Problem Comparisons by Sample Size.

Size	**Goals**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
20	**Goal 1**	114.49	113.10	108.68	114.39	108.47
	Goal 2	23.53	40.94	37.29	34.98	41.76
	Goal 3	3.97	4.18	3.08	5.38	3.67
	Goal 4	39.97	39.98	38.58	41.38	39.67
40	**Goal 1**	123.56	122.12	119.57	122.55	116.85
	Goal 2	37.04	45.22	41.22	38.60	45.00
	Goal 3	5.58	4.26	3.37	5.51	3.79
	Goal 4	41.58	40.15	39.14	41.51	39.79
60	**Goal 1**	128.98	127.92	123.40	127.65	121.08
	Goal 2	36.16	43.53	40.82	37.78	43.83
	Goal 3	5.64	4.30	3.37	5.65	3.90
	Goal 4	41.64	40.23	38.90	41.65	39.90

shown in Figure 5.5a, since the goals are ordered lexicographically, and the run time values do not change dramatically among methods, MILP & MILP is superior to the other methods for sample size 20. However, it is observed that MILP & MILP underperformed in run time in sample size 40 and size 60 charts which are shown in Figure 5.5b and Figure 5.5c, respectively.

In fact, with each dynamism level increase, the model tries to add two more emergent patients to the route. Since the priority value of emergent patients is also 5, we expect to see an increase of about 10 for each dynamism level increase on average. However, as seen in Table 5.15, the increases observed between six and eight in the Goal 1 values of MILP & GRASP+VNS, GRASP+VNS & MILP and GRASP+VNS & HYBRID methods. Here the following deductions can be made. Since emergent patients have to be included in the route, these methods postponed more than other methods in order to include emergent patients in the routes. It can be said that methods, which have both a low Goal 1 value and a high Goal 4 value,

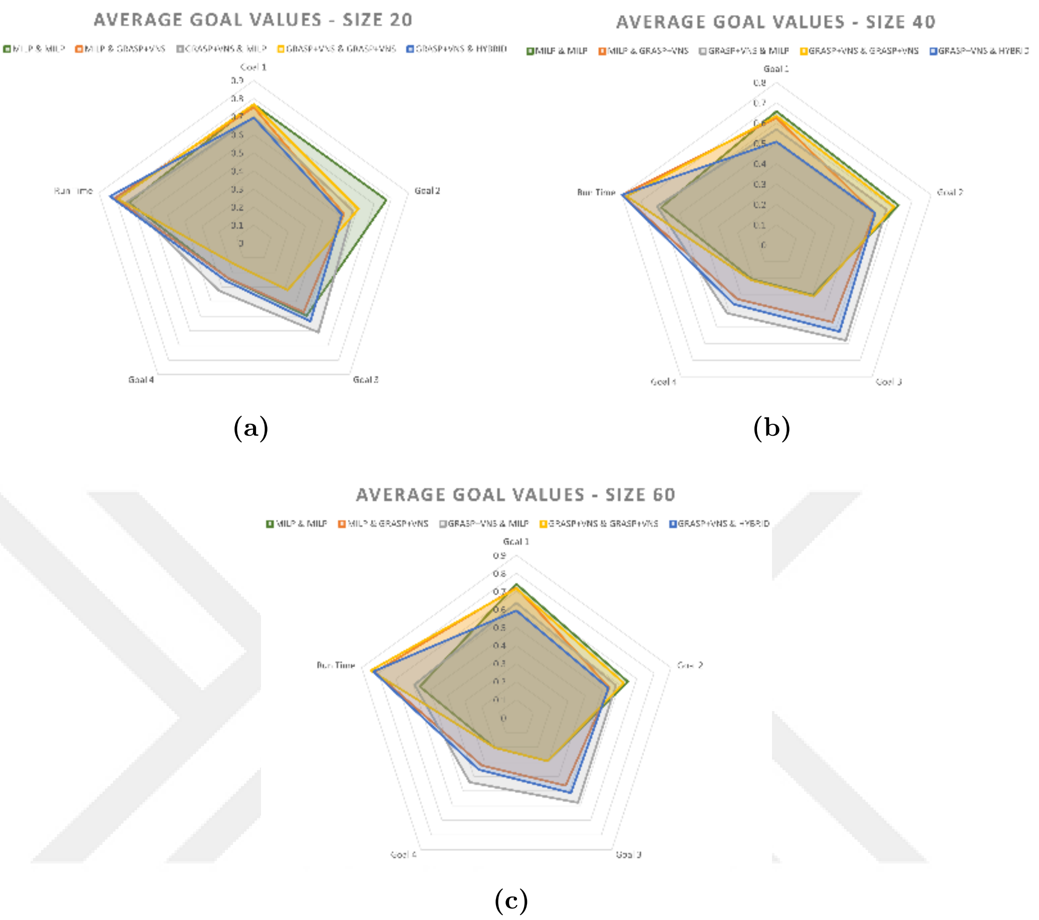


Figure 5.5: The Comparison of Methods According to Sample Size.: (a) Sample Size 20, (b) Sample Size 40 (c) Sample Size 60.

cannot perform well. The statistical analysis of the goal values by dynamism level is compared, and the related table is available in Appendix B, Table B.2.

While examining the data that we present in Table 5.12 together with the run time data in Figure 5.15, it is observed that MILP & MILP and GRASP+VNS & GRASP+VNS give close results and GRASP+VNS & GRASP+VNS are quite successful in run time. According to Figure 5.6c, it has been observed that MILP & MILP is far more successful than all methods in Goal 2.

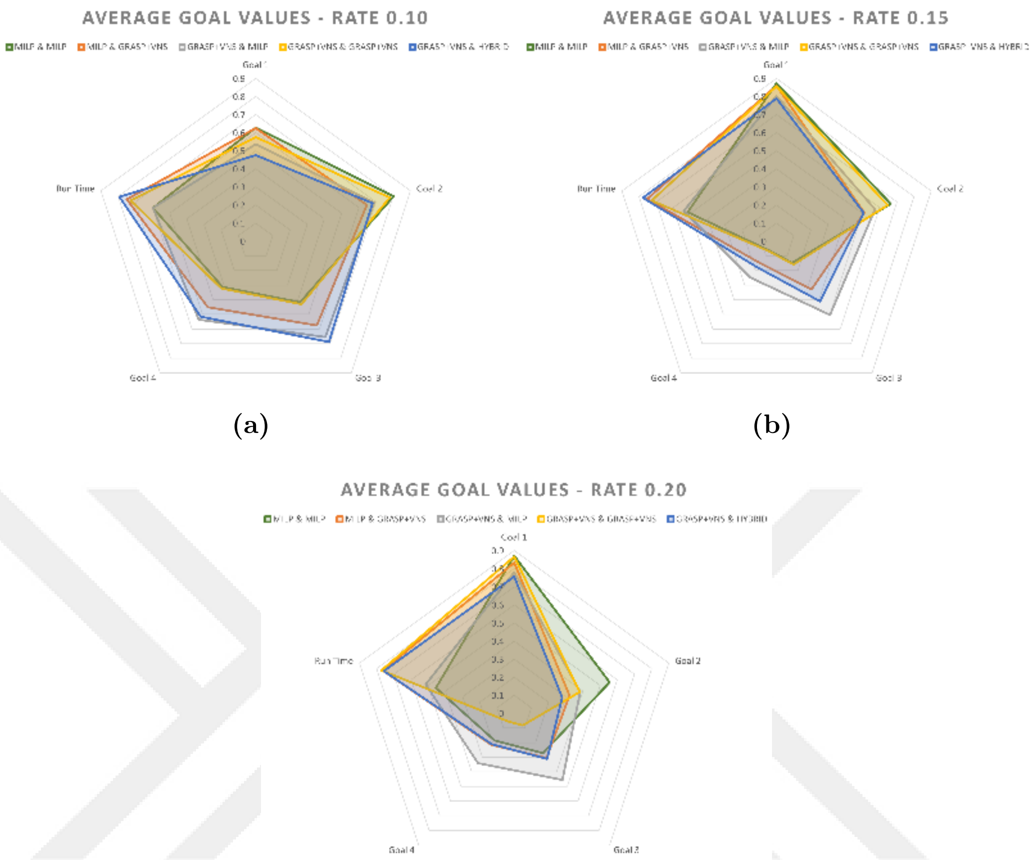
As mentioned in the previous paragraphs, it is not possible to fully evaluate the models only by means of the goals. When comparing the methods, it is necessary to examine the performance metrics that are not included in the goals but are indirectly

Table 5.15: Simulation Results: Dynamic Problem Comparisons by Sample Dynamism Level.

Dynamism Level	Goals	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID
Low	**Goal 1**	112.72	112.64	109.70	110.96	107.68
	Goal 2	21.36	29.49	27.29	22.53	27.80
	Goal 3	3.73	2.71	2.21	3.64	1.99
	Goal 4	39.73	38.54	37.81	39.64	37.99
Mid	**Goal 1**	122.82	121.79	117.24	121.58	116.12
	Goal 2	35.75	43.09	40.16	36.61	43.23
	Goal 3	5.66	4.45	3.29	5.58	3.91
	Goal 4	41.66	40.34	38.89	41.58	39.91
High	**Goal 1**	131.91	128.62	124.14	131.52	122.06
	Goal 2	39.90	56.99	52.48	52.66	60.09
	Goal 3	5.83	5.56	4.37	7.36	5.52
	Goal 4	41.83	41.47	39.99	43.36	41.52

affected. In Table 5.16, there are four performance metrics to use in the evaluation of the methods. The first is the “Total Run Time” metric, which shows how long the method takes to place the emergent patients on the route during a day, and is in seconds. The “Postponements” metric displays how many people on the first route are postponed at the end of the day. The “HSP Changes” metric represents the number of times patients have changed their HSP by the end of the day. Also, the statistical analysis of the results grouped by size and dynamism level are given in Appendix B in Tables B.4 and B.5, respectively.

When Table 5.16 is examined, it is observed that MILP & MILP is the worst method in terms of run time and GRASP+VNS & HYBRID is the best method. However, the information that will give the real insight here is that the GRASP+VNS & GRASP+VNS method approaches to be the fastest solving method as the size and dynamism level increases, that is, as the problem becomes more difficult. It



(a) (b) (c)

Figure 5.6: The Comparison of Methods According to Sample Dynamism Level.: (a) Low-Level Dynamism Sample, (b) Mid-Level Dynamism Sample, (c) High-Level Dynamism Sample.

even surpasses the hybrid method in 60-High samples. The statistical analysis of the performance metric values obtained from our proposed methods is available in Appendix B, Table B.3. As shown in Figure 5.4, it returns results close to MILP & MILP results in goal values. For this reason, GRASP+VNS & GRASP+VNS stands out as an ideal method for those who want both speed and solution quality.

MILP & MILP and GRASP+VNS & MILP methods where dynamic part is solved with MILP in the postponement results have been very successful. This is because Goal 1 in LGP method is the total priority. When the MILP model achieves an optimal solution, naturally postponement occurs at the lowest level. Methods

Table 5.16: The Evaluation of the Methods Used in Dynamic Problem through Performance Metrics.

Size	Dynamism Level	Performance Metrics	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID
20	Low	Total Run Time	73.67	61.14	83.36	71.13	48.61
		# Postponements	0.00	0.00	0.00	0.13	0.93
		# HSP Changes	1.17	2.20	1.53	2.27	1.67
	Mid	Total Run Time	100.85	74.57	97.51	91.63	58.12
		# Postponements	0.00	0.48	0.00	0.97	1.39
		# HSP Changes	2.57	5.70	2.77	3.63	3.13
	High	Total Run Time	207.32	132.24	193.84	134.02	123.37
		# Postponements	0.47	0.80	0.04	2.57	2.14
		# HSP Changes	1.40	11.30	4.20	7.90	4.50
40	Low	Total Run Time	121.35	71.61	115.61	76.89	61.84
		# Postponements	0.00	0.04	0.00	0.07	0.67
		# HSP Changes	2.33	5.83	2.90	5.83	4.03
	Mid	Total Run Time	195.12	106.50	207.67	122.68	104.17
		# Postponements	0.00	0.00	0.00	0.72	1.21
		# HSP Changes	3.57	10.63	3.53	8.30	5.40
	High	Total Run Time	293.90	133.90	246.69	145.28	142.28
		# Postponements	0.12	1.08	0.04	0.74	2.31
		# HSP Changes	5.30	15.83	5.20	14.30	7.43
60	Low	Total Run Time	97.50	49.75	93.34	46.86	40.59
		# Postponements	0.00	0.04	0.00	0.50	0.76
		# HSP Changes	2.50	8.53	2.63	5.27	3.73
	Mid	Total Run Time	235.78	93.31	207.72	88.35	86.81
		# Postponements	0.00	0.07	0.00	0.48	1.19
		# HSP Changes	3.93	14.97	5.37	12.97	7.33
	High	Total Run Time	324.35	126.11	294.78	100.24	134.04
		# Postponements	0.15	0.70	0.08	0.96	2.08
		# HSP Changes	6.17	23.57	7.37	19.27	7.80

using GRASP+VNS metaheuristic algorithms also have acceptable postponement values. However, the Hybrid method is unsuccessful in postponement results. It has also been proven that the Hybrid method's postponement values will be higher based on the previous Goal 1 values.

It is seen that the models that solve the dynamic problem with the GRASP+VNS metaheuristic algorithm are higher in HSP Change numbers. In particular, the GRASP+VNS & GRASP+VNS metaheuristic changes planned HSPs of the patients and reaches the priority value close to MILP & MILP but faster than it. In fact, it has reached a solution 2-3 times faster in difficult cases.

5.8 The Effect of HSPs' Cooperation

The cooperation of the HSPs, which is determined as the main strategy in this thesis, will be analysed in this section. In the case of cooperation, a patient may be visited by another HSP other than the HSP to which he is originally assigned. However, in the currently applied methods, if a patient is assigned to an HSP, only that HSP can be involved. This situation has been named as the *Non-Cooperation* strategy in our study.

First of all, when planning the initial route to make this analysis, it was decided to use samples with size 40 patients, since it is medium-sized. Since our strategy mainly concerns the dynamic problem, samples were taken from three different dynamism levels. Samples 40-Low-7, 40-Mid-0, and 40-High-8 were randomly selected. GRASP+VNS & HYBRID method was used in the solution. There is an one-minute time limit for the CPLEX and GRASP+VNS metaheuristic algorithms and they were run in parallel on 28 cores.

Table 5.17 is included in order to analyse the contribution of the cooperation strategy to the dynamic problem. A total of 18 emergent patients come from three different samples mentioned above. Each row shows the status after the patient in the first column is included in the route and schedule of the HSPs.

Evaluating the Goal 1 and Postponement values at Table 5.17, it is observed that the Cooperation strategy changed the HSPs of the patients in order to serve more patients and collect more priorities. However, when the Goal 2 value is considered, serving more patients forced the model to change arrival times of the patients. The reason why the same results cannot be obtained in case of no HSP change is due to the fact that the approaches were operated as Hybrid which includes metaheuristic algorithm.

Table 5.18 shows the average values according to the samples. The detailed statistical analysis of theses comparison values according to the dynamism levels is

Table 5.17: The Comparison of Cooperation and Non-Cooperation by Goals, Run Time, and Postponements

Patient	Sample	Non-Cooperation					Cooperation					
		Goal 1	Goal 2	Goal 3	Goal 4	# Post.	Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes
1	40-Low-7	100	10.23	1.02	35.41	0	100	10.54	0	34.88	0	1
2		102	4.2	0.45	35.25	1	105	18.36	0.61	36.61	0	4
3		107	10.21	1.06	37.06	2	110	26.88	1.63	37.63	0	1
4		109	11.5	1.16	37.16	2	112	35.08	2.24	38.24	1	4
5	40-Mid-0	97	7.64	0.89	36.34	0	102	10.88	0.18	35.08	0	1
6		102	10.88	1.25	37.25	0	107	19.27	0.39	36.03	0	0
7		107	13.79	2.25	38.25	0	108	27.80	1.26	36.89	1	1
8		109	13.9	2.33	38.33	1	113	35.98	2.33	37.95	1	1
9		111	15.8	2.43	38.43	2	118	42.03	3.16	38.77	1	1
10		114	15.76	2.29	38.29	3	123	47.09	4.07	39.67	1	0
11	40-High-8	101	8.86	1.14	36.9	0	101	11.63	0	34.96	0	0
12		103	7.07	0.87	36.63	1	106	21.89	0.81	36.23	0	1
13		108	14.74	1.92	37.92	1	111	31.27	1.88	37.20	0	3
14		111	15.53	1.87	37.87	2	116	39.72	3.45	39.05	0	1
15		113	14.68	1.28	37.28	3	121	45.58	3.49	39.09	0	2
16		118	15.72	2.35	38.35	3	126	53.20	4.55	39.33	0	0
17		121	12.29	2.88	38.88	3	129	59.10	5.52	39.45	1	0
18		123	13.71	3.14	39.14	4	126	62.22	4.07	39.97	2	2

Table 5.18: The Comparison of Cooperation and Non-Cooperation by Samples' Averages of Goals, Postponements, and the Number of HSP Changes

Sample	Non-Cooperation					Cooperation					
	Goal 1	Goal 2	Goal 3	Goal 4	# Post.	Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes
40-Low-7	104.50	9.04	0.92	36.22	1.25	106.75	22.72	1.12	36.84	0.25	2.50
40-Mid-0	106.67	12.96	1.91	37.82	1.00	111.83	30.51	1.90	37.40	0.67	0.67
40-High-8	112.44	13.15	1.97	37.92	2.22	117.67	41.30	2.97	38.33	0.44	1.00

available in Appendix B, Table B.6. As seen in Table 5.18, there are no correlations between dynamism levels and HSP changes. As the dynamism level increases, goal values and postponement values increase as expected. Although the increase is in both strategies, the difference in Goal 1 value between them increases as the dynamism level increases. Table 5.18 shows that the cooperation strategy yields better results under every dynamic condition.

5.9 The Analysis of 3-HSPs' Cooperation

Until this section, all analysis has shown the results of the instances with two-HSPs. In order to show the contributions of a new HSP in the system, we solved the static and dynamic problems for a new third HSP with the Hybrid approach. We call this version as 3-HSPs' Cooperation. In this section, we analyze the comparison between 2-HSPs' Cooperation and 3-HSPs' Cooperation with the same emergent patients.

Table 5.19: The Comparison of 2-HSPs' Cooperation and 3-HSPs' Cooperation by Goals, Postponements, and the Number of HSP Changes

Patient	Sample	2-HSPs' Cooperation						3-HSPs' Cooperation					
		Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes	Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes
1	40-Low-7	100	10.54	0	34.88	0	1	136	16.17	1.60	49.29	0	4
2		105	18.36	0.61	36.61	0	4	141	6.12	2.08	50.08	0	1
3		110	26.88	1.63	37.63	0	1	146	9.50	2.35	50.35	0	3
4		112	35.08	2.24	38.24	1	4	151	11.90	3.05	51.05	0	0
5	40-Mid-0	102	10.88	0.18	35.08	0	1	135	7.89	0.62	48.29	0	2
6		107	19.27	0.39	36.03	0	0	140	6.73	1.06	49.06	0	1
7		108	27.80	1.26	36.89	1	1	145	10.35	1.87	49.87	0	1
8		113	35.98	2.33	37.95	1	1	150	9.45	1.87	49.87	0	6
9		118	42.03	3.16	38.77	1	1	155	12.64	2.54	50.54	0	1
10		123	47.09	4.07	39.67	1	0	160	15.04	3.12	51.12	0	0
11	40-High-8	101	11.63	0	34.96	0	0	137	17.58	1.45	49.45	0	4
12		106	21.89	0.81	36.23	0	1	142	8.01	2.75	50.75	0	3
13		111	31.27	1.88	37.20	0	3	147	7.68	2.81	50.81	0	5
14		116	39.72	3.45	39.05	0	1	152	7.83	3.51	51.51	0	3
15		121	45.58	3.49	39.09	0	2	157	8.97	4.35	52.35	0	0
16		126	53.20	4.55	39.33	0	0	162	13.95	4.71	52.71	0	3
17		129	59.10	5.52	39.45	1	0	167	16.36	5.62	53.62	0	2
18		126	62.22	4.07	39.97	2	2	170	17.46	6.05	54.05	1	2

In Table 5.19, 3-HSPs' Cooperation has more HSP changes and less postponed patients than 2-HSPs' Cooperation in the same datasets. As expected, while increasing the HSP number, the cooperation effect increases and the solution reaches more collected priority values. For comparing the collected priority values, looking at the

Goal-1 column can be misleading because it includes the priority values of the third HSP's initially planned patients. We have deduced this comparison by observing the " Post." column. Also, the arrival time changes of 3-HSPs' Cooperation are up to eight times smaller than 2-HSPs' Cooperation, although 3-HSPs' Cooperation's values are calculated with all patients of three HSPs.

In Table 5.19, 3-HSPs' Cooperation has more HSP changes and less postponed patients than 2-HSPs' Cooperation in the same datasets. As expected, while increasing the HSP number, the cooperation effect increases and the solution reaches more collected priority values. For comparing the collected priority values, looking at the Goal-1 column can be misleading because it includes the priority values of the third HSP's initially planned patients. We have deduced this comparison by looking " Postponements" column. Also, the arrival time changes of 3-HSPs' Cooperation are up to eight times smaller than 2-HSPs' Cooperation, although 3-HSPs' Cooperation's values are calculated with all patients of three HSPs. When the average performance metrics of the instances is examined in the Table 5.20, the superiority of 3-HSPs' Cooperation in the "Goal-2" and " Postponements" columns is clearly seen. Of course, the superiority of 3-HSPs in the first two goals causes a decrease in Goal-3 and Goal-4 values, which represent overtime and total time, respectively. Including the new patient in the plan without postponing will increase the total time and thus the overtime as expected. However, since we apply the LGP methodology, the superiority in Goal-1 and Goal-2 clearly proves that 3-HSPs' Cooperation reaches better solutions than 2-HSPs' Cooperation.

Table 5.20: The Comparison of 2-HSPs' Cooperation and 3-HSPs' Cooperation by Instances' Averages of Goals, Postponements, and the Number of HSP Changes

Sample	2-HSPs' Cooperation						3-HSPs' Cooperation					
	Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes	Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes
40-Low-7	106.75	22.72	1.12	36.84	0.25	2.50	143.50	10.92	2.27	50.19	0	2
40-Mid-0	111.83	30.51	1.90	37.40	0.67	0.67	146.50	10.35	1.85	51.46	0	1.84
40-High-8	117.67	41.30	2.97	38.33	0.44	1.00	154.25	12.23	3.91	51.91	0.13	2.75

We also examine the results of the 3-HSPs' Cooperation with new emergent patients. These new emergent patients are coming with added third HSP according to dynamism level. Due to the fact that each HSP has its own region and patient cluster, each HSP should have its own emergent patients. The result and summary tables are added to the Appendix B, B.7 and B.8.

Chapter 6

CONCLUSIONS

In this study, we have focused on home healthcare services and considered the inclusion of dynamically occurring emergent patients in daily plans of the HSPs that are prepared according to the priorities of the patients. As the emergent cases that come dynamically in the problem are included in the daily plans, it is aimed not to postpone the highest prioritized patients, to ensure care continuity as much as possible, to make the least possible changes in the arrival times of the patients who have been already scheduled in the initial plan, and to minimize the total overtime duration of the HSPs. In addition to these objectives, in order to make the problem more realistic, several constraints have been posed, such as limited working hours and overtime duration, servicing each emergent patient within two hours after his/her request arrives, and not allowing the postponement of the visits of patients with highest priority. In addition, time-dependent travel times between the locations have been defined for a more realistic setting. More importantly, we have incorporated the coordination of the HSPs into the problem for the first time in the literature in order to utilize total service capacity better by a centralized optimization approach that generates the plans of all HSPs simultaneously.

We define a static and a dynamic problem. In the dynamic problem, emergent patients are inserted into the initial plan in accordance with the constraints and objectives described above. In order to create the initial plan used in the dynamic problem, it is necessary to solve a problem that has similar goals and constraints as the dynamic problem. We call this problem as the static problem. The static problem generates an HSP's single day schedule and route consisting of visits to only patients assigned to the HSP a priori. Unlike the dynamic problem, the patients to be visited are known at the beginning of the day, and no overtime is allowed. The static problem aims to include patients with higher priority in the initial plan and to obtain minimum total travel time. The initial plan obtained as a result of solving this problem is then used as an input to the dynamic problem.

In this study, we have taken the lexicographic goal programming (LGP) approach because our goals of both the static and dynamic problems have different priority levels. In the LGP approach, we solve the problem for each goal in a lexicographic order. After the solution of a problem with a higher level goal is obtained, adding the previously obtained optimal goal values are required to be obtained in the new problem by means of adding an equality constraint. This process continues for each level goal until the last and least prioritized goal. In the lexicographic order, our goals in the static problem are total priority maximization, and total travel time minimization. On the other hand, in the dynamic problem we have four goals. Our goals in the dynamic problem in the lexicographic order are total priority maximization, arrival time changes minimization, total overtime minimization, and total time minimization. The first goal in this lexicographic order, that is the total priority maximization, refers to maximizing the total priority values of the visited patients. Arrival time change is defined by means of the difference between the initially planned and finally realized visit times. Since the schedule changes throughout the day are undesirable for the patients, this goal is to be minimized. Total overtime minimization has more importance than the total travel time minimization because overtime is costly, a creates a burden and infection risk for the HSPs. Thus, we prioritized the overtime instead of the total time.

This thesis mainly focuses on solving the dynamic problem efficiently, but our study also proposes solution methodology for the static problem since its solution is needed as an input to the dynamic problem. In order to include the emergent patients in the routes and to reach better objective function values in the dynamic problem, HSPs are allowed to switch their patients, if that brings an advantage. However, we give priority to the HSP-patient assignments for patients in the HSPs group to ensure care continuity as much as possible. We have referred to this strategy as cooperation in our study.

As for methodology, first, we have developed a MILP model and a GRASP+VNS metaheuristic for the static problem. For the dynamic problem, five different methods have been proposed: MILP & MILP, MILP & GRASP+VNS, GRASP+VNS & MILP, GRASP+VNS & GRASP+VNS, GRASP+VNS & HYBRID. Here, we state first the solution method for the static problem and secondly the dynamic problem.

In the static problem, two solutions method are developed: MILP model for an exact solution and GRASP+VNS metaheuristic. These two solutions methods are tested for their performance and comparing each other a modified COVID-19 case data.

In the dynamic problem, three solutions methods are developed MILP model, GRASP+VNS metaheuristic and Hybrid approach. Since the dynamic problem methods use the outputs of the static problem, it is important which method is used in the static problem. Therefore, these three solutions methods are combined with the methods of the static problem as following: MILP & MILP, MILP& GRASP + VNS, GRASP + VNS& MILP, GRASP + VNS& GRASP + VNS, GRASP + VNS & HYBRID. The first part of the compound names indicates the methods used in the static problem, and the second part is the method of the dynamic problem. These five approaches are tested and compared among them computationally by using the same data with the static problem and the outputs of the static problem.

Our data used in the computational study includes patient locations, estimated treatment duration, priorities for HSPs. This data has been modified from our previous study on COVID-19. Patient locations were generated from a district of Istanbul. There are 30 instances for each patient size (20, 40, 60) and for each dynamism level (low, mid, high). Here the sample size refers to the number of patients in the static problem. Levels of low, mid, and high indicate that four, six, and eight emergent patients arrive at randomly times in the dynamic problem, respectively. The dynamic problem uses the result of the outputs of the static problem as the initial plan. In the dynamic problem, the model re-optimizes the HSPs' routes when each emergent patient arrives.

Computational study details are mentioned the following. First of all, grid search was performed with our data for parameter hyper-tuning of the GRASP and VNS heuristics. For the static problem, we selected 0.5 as restricted list candidate (RCL) threshold of GRASP, 50 as iteration limit of GRASP, 50 as non-improving iteration limit of GRASP, 100 as non-improving iteration limit of VNS. For the static problem, the values of the same parameters are selected 0.7, 100, 10, 100 respectively. Hypertuning is done separately for both of the static and dynamic problems. Also, we designed two different VNS variants, which are VNS-1 and VNS-2, by arranging 2-opts, 2-Exchange and One Add-One Drop algorithms in different ways, and we tested these VNS versions. VNS-2 version with this arrangement "One add-One drop-2-Exchange-2 Opt" is selected. After improving the GRASP and VNS algorithms, outputs for the static problem were taken from both our GRASP+VNS heuristic and our MILP model data. Based on the initial plans taken in the static problem, the dynamic problem outputs were obtained with the five models we proposed. We ran all our models in parallel.

We have observed the following findings from our computational experiments. For the static problem, the MILP model has an advantage over the GRASP+VNS metaheuristic algorithm in terms of lexicographically ordered goals. However, the GRASP+VNS metaheuristic algorithm provides 18.4 times faster solutions on the average over the instances. Moreover, the GRASP+VNS metaheuristic algorithm is better than the MILP model in 3% of the solutions and has obtained the same solution as the MILP in 38% of the solutions. That shows the success of the metaheuristic. Also, in solutions where MILP is better than the GRASP+VNS, the GRASP+VNS achieves achieves an average value with 4.01% GAP from the MILP's solution.

For the dynamic problem, we have five different combined methods: MILP & MILP, MILP & GRASP + VNS, GRASP + VNS & MILP, GRASP + VNS & GRASP + VNS, GRASP + VNS & HYBRID. When the outputs of the dynamic problem checks, we can observe that MILP & MILP and GRASP+VNS & GRASP+VNS methods stand out in terms of both run time and solution quality. The fact that the designed hybrid method returns the worst results always shows that a greedy approach is not suitable and should not be preferred. If there is a better GRASP+VNS solution instead of the MILP model solution, which cannot provide an optimal solution due to the hybrid model design, hybrid model used it. Instances in which MILP & MILP solution is better than GRASP+VNS &

GRASP+VNS solution have 1.87% gap among them for Goal 1. At the same time, in point of run time perspective, GRASP+VNS & GRASP+VNS solution found a solution up to 3.24 times faster as sample size and dynamism level values increased. Similar to the static problem, GRASP+VNS & GRASP+VNS solution is a good alternative to the MILP & MILP solution that can reach similar quality solutions faster. GRASP+VNS & MILP and MILP & GRASP+VNS solutions, which are other solution methods, generally showed average performance in performance evaluations.

An analysis has also been performed to show the benefits from our proposed strategy, namely HSPs' cooperation. For the analysis, a system that did not allow cooperation has been established for benchmarking purpose and solved as a dynamic problem. This system has been referred to as "Non-Cooperation" in short. We have observed that more priority values were collected with the HSP changes at the patient visits made in the cooperation strategy, fewer patients were postponed, and therefore more patients were served. To express it numerically; on the same instances, an average of 3% better results were obtained as the total priority, and 3.54 times more patients were postponed in the non-cooperative strategy. However, the arrival time change, overtime and total route duration values have deteriorated in the cooperation strategy because serving more patients triggers longer work time and the schedule changes. However, this may be acceptable since getting a better score in Goal 1 has priority over the other goals in our approach. Also, no worse solution was obtained than the non-cooperation strategy in terms of total priority value (Goal 1) in all solutions. In order to show the effect of the HSP numbers on the cooperation strategy, a third HSP is added to the dataset, and the GRASP+VNS & HYBRID method is solved again as the 3-HSPs' Cooperation. When the results are compared between 2-HSPs' cooperation and 3-HSPs' cooperation, it is seen that 3-HSPs' cooperation can reach fewer postponements by using the cooperation strategy.

The most salient feature of our study is the use of the cooperation strategy in a dynamic setting. Home healthcare routing and scheduling has been studied widely in the literature where many problem variants with different characteristics have been addressed. In fact, there are home healthcare routing and scheduling problem studies addressing the features that have been addressed in our study, such as prioritization of the patients, continuity of care requirement, HSPs' cooperation, time-dependent travel times, minimization of the total route time, minimization of the arrival time changes and the emergent patients which must be visited in two hours. However, in our literature review we have not encountered any study on the dynamic problem that includes all these features at the same time. Furthermore, lexicographic goal programming approach is also novel for home healthcare studies. Finally, we point out some future work directions as follows.

- Several studies in the literature [Shahnejat-Bushehri et al., 2019, Liu et al., 2021] require two or more HSPs to deal with some special patients at the same time and this feature is referred to as synchronization. As an extension of our work, bot the cooperation strategy and the synchronization feature can be considered in the problem.

- If the long-term treatment process of the patient is considered in a multiple day setting with some varying parameters over time and care continuity is considered as an important issue that can be achieved over time, a generalization of this study with multiple-periods can focus on a longer planning horizon. In such a setting, for instance, patient priorities may also be updated over time according to their changing health conditions and treatment urgency.

- The type of equipment required for the treatment of each patient and their availability can be explicitly considered since the number of available equipment of different types may be limited and may need to be shared among the HSPs at different times. In addition, in a generalization of this study with a weekly or monthly planning horizon, the amount of equipment to be acquired for the medium and long term for better service quality can be a tactical decision in the problem.

- The methods developed in this study can be incorporated into a remote treatment software, which has become widespread nowadays, to form a decision support system. There may be further features in such a system such as keeping track of suitable times for the physician/nurse who will provide remote treatment. In addition, patients can be given time windows and the arrival time to the patients can be tracked.

- With smart watches, patients' health data can be accumulated and checked in real-time. As a result, these data can be utilized in machine learning models to predict any future urgency situation, that is the candidate emergent patients. Then, we can minimize the distance of the candidate emergent patient to the patients on the route. Therefore, if a call comes from the candidate emergent patient, the waiting time will be reduced.

BIBLIOGRAPHY

[Ait Haddadene et al., 2019] Ait Haddadene, S. R., Labadie, N., and Prodhon, C. (2019). Bicriteria vehicle routing problem with preferences and timing constraints in home health care services. *Algorithms*, 12(8).

[Bennett and Erera, 2011] Bennett, A. R. and Erera, A. L. (2011). Dynamic periodic fixed appointment scheduling for home health. *IIE Transactions on Healthcare Systems Engineering*, 1(1):6–19.

[Bowers et al., 2015] Bowers, J., Cheyne, H., Mould, G., and Page, M. (2015). Continuity of care in community midwifery. *Health Care Management Science*, 18(2):195–204.

[Cappanera and Scutellà, 2013] Cappanera, P. and Scutellà, M. G. (2013). Home care optimization: impact of pattern generation policies on scheduling and routing decisions. *Electronic Notes in Discrete Mathematics*, 41:53–60.

[Chaieb et al., 2020] Chaieb, M., Jemai, J., and Mellouli, K. (2020). A decomposition - construction approach for solving the home health care scheduling problem. *Health Care Management Science*, 23(2):264–286.

[Cinar et al., 2022] Cinar, A., Salman, F. S., Araz, O. M., and Parcaoglu, M. (2022). Managing home healthcare services with dynamic arrivals during a public health emergency.

[Cinar et al., 2021] Cinar, A., Salman, F. S., and Bozkaya, B. (2021). Prioritized single nurse routing and scheduling for home healthcare services. *European Journal of Operational Research*, 289(3):867–878.

[Decerle et al., 2017] Decerle, J., Grunder, O., Hajjam El Hassani, A., and Barakat, O. (2017). A general model for the home health care routing and scheduling problem with route balancing. *IFAC-PapersOnLine*, 50(1):14662–14667. 20th IFAC World Congress.

[Decerle et al., 2018a] Decerle, J., Grunder, O., Hajjam El Hassani, A., and Barakat, O. (2018a). A memetic algorithm for a home health care routing and scheduling problem. *Operations Research for Health Care*, 16:59–71.

[Decerle et al., 2019] Decerle, J., Grunder, O., Hajjam El Hassani, A., and Barakat, O. (2019). A hybrid memetic-ant colony optimization algorithm for the home health care problem with time window, synchronization and working time balancing. *Swarm and Evolutionary Computation*, 46:171–183.

[Decerle et al., 2018b] Decerle, J., Grunder, O., Hassani, A. H. E., and Barakat, O. (2018b). Impact of the workload definition on the multi-objective home health care problem. *IFAC-PapersOnLine*, 51(11):346–351. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[Demirbilek et al., 2019] Demirbilek, M., Branke, J., and Strauss, A. (2019). Dynamically accepting and scheduling patients for home healthcare. *Health Care Management Science*, 22(1):140–155.

[Demirbilek et al., 2021] Demirbilek, M., Branke, J., and Strauss, A. K. (2021). Home healthcare routing and scheduling of multiple nurses in a dynamic environment. *Flexible Services and Manufacturing Journal*, 33(1):253–280.

[Dengiz et al., 2020] Dengiz, A. O., Atalay, K. D., and Altiparmak, F. (2020). Multiple service home health care routing and scheduling problem: A mathematical model. In Karwowski, W., Trzcielinski, S., and Mrugalska, B., editors, *Advances in Manufacturing, Production Management and Process Control*, pages 289–298. Springer International Publishing, Cham.

[Di Mascolo et al., 2014] Di Mascolo, M., Espinouse, M.-L., and Ozkan, C. E. (2014). Synchronization between human resources in home health care context. In Matta, A., Li, J., Sahin, E., Lanzarone, E., and Fowler, J., editors, *Proceedings of the International Conference on Health Care Systems Engineering*, pages 73–86. Springer International Publishing, Cham.

[Du et al., 2019] Du, G., Zheng, L., and Ouyang, X. (2019). Real-time scheduling optimization considering the unexpected events in home health care. *Journal of Combinatorial Optimization*, 37(1):196–220.

[Falvey et al., 2020] Falvey, J. R., Krafft, C., and Kornetti, D. (2020). The essential role of home-and community-based physical therapists during the covid-19 pandemic. *Physical therapy*, 100(7):1058–1061.

[Fathollahi-Fard et al., 2021] Fathollahi-Fard, A. M., Ahmadi, A., and Karimi, B. (2021). Multi-objective optimization of home healthcare with working-time balancing and care continuity. *Sustainability*, 13(22).

[Fikar and Hirsch, 2015] Fikar, C. and Hirsch, P. (2015). A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production*, 105:300–310. Decision-support models and tools for helping to make real progress to more sustainable societies.

[Fikar and Hirsch, 2018] Fikar, C. and Hirsch, P. (2018). Evaluation of trip and car sharing concepts for home health care services. *Flexible Services and Manufacturing Journal*, 30(1):78–97.

[Fikar et al., 2016] Fikar, C., Juan, A. A., Martinez, E., and Hirsch, P. (2016). A discrete-event driven metaheuristic for dynamic home service routing with synchronised trip sharing. *European Journal of Industrial Engineering*, 10(3):323–340. PMID: 76382.

[Grenouilleau et al., 2019] Grenouilleau, F., Legrain, A., Lahrichi, N., and Rousseau, L.-M. (2019). A set partitioning heuristic for the home health care routing and scheduling problem. *European Journal of Operational Research*, 275(1):295–303.

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

[Hassen et al., 2019] Hassen, H. B., Tounsi, J., and Bachouch, R. B. (2019). An artificial immune algorithm for hhc planning based on multi-agent system. *Procedia Computer Science*, 164:251–256. CENTERIS 2019 - International Conference on ENTERprise Information Systems / ProjMAN 2019 - International Conference on Project MANagement / HCist 2019 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2019.

[He and Shi, 2019] He, C. and Shi, Y. (2019). Workforce scheduling and routing for home health care services. In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, CSAE 2019. Association for Computing Machinery, New York, NY, USA.

[Heching and Hooker, 2016] Heching, A. and Hooker, J. N. (2016). Scheduling home hospice care with logic-based benders decomposition. In Quimper, C.-G., editor, *Integration of AI and OR Techniques in Constraint Programming*, pages 187–197. Springer International Publishing, Cham.

[Heching et al., 2019] Heching, A., Hooker, J. N., and Kimura, R. (2019). A logic-based benders approach to home healthcare delivery. *Transportation Science*, 53(2):510–522.

[Hertz and Lahrichi, 2009] Hertz, A. and Lahrichi, N. (2009). A patient assignment algorithm for home care services. *Journal of the Operational Research Society*, 60(4):481–495.

[Hosseinpour-Sarkarizi et al., 2020] Hosseinpour-Sarkarizi, A., Davari-Ardakani, H., and Izadbakhsh, H. (2020). Home health care routing and scheduling problem under uncertainty considering patient preferences and service desirability. *Scientia Iranica*.

[Lin et al., 2021] Lin, M., Ma, L., and Ying, C. (2021). Matching daily home health-care demands with supply in service-sharing platforms. *Transportation Research Part E: Logistics and Transportation Review*, 145:102177.

[Liu et al., 2019a] Liu, R., Tao, Y., and Xie, X. (2019a). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. *Computers Operations Research*, 101:250–262.

[Liu et al., 2017] Liu, R., Yuan, B., and Jiang, Z. (2017). Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements. *International Journal of Production Research*, 55(2):558–575.

[Liu et al., 2019b] Liu, R., Yuan, B., and Jiang, Z. (2019b). A branch-and-price algorithm for the home-caregiver scheduling and routing problem with stochastic travel and service times. *Flexible Services and Manufacturing Journal*, 31(4):989–1011.

[Liu et al., 2021] Liu, W., Dridi, M., Fei, H., and El Hassani, A. H. (2021). Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. *Expert Systems with Applications*, 183:115307.

[López-Santana et al., 2016] López-Santana, E. R., Espejo-Díaz, J. A., and Méndez-Giraldo, G. A. (2016). Multi-agent approach for solving the dynamic home health care routing problem. In Figueroa-García, J. C., López-Santana, E. R., and Ferro-Escobar, R., editors, *Applied Computer Sciences in Engineering*, pages 188–200. Springer International Publishing, Cham.

[Ly Nguyen and Montemanni, 2016] Ly Nguyen, T. V. and Montemanni, R. (2016). Integrated home health care optimization via genetic algorithms and mathematical programming. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 553–561.

[Malagodi et al., 2021] Malagodi, L., Lanzarone, E., and Matta, A. (2021). Home care vehicle routing problem with chargeable overtime and strict and soft preference matching. *Health Care Management Science*, 24(1):140–159.

[Manerba and Mansini, 2016] Manerba, D. and Mansini, R. (2016). The nurse routing problem with workload constraints and incompatible services. *IFAC-PapersOnLine*, 49(12):1192–1197. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016.

[Mankowska, 2016] Mankowska, D. S. (2016). Synchronization in vehicle routing: Benders' decomposition for the home health care routing and scheduling problem. In Kotzab, H., Pannek, J., and Thoben, K.-D., editors, *Dynamics in Logistics*, pages 159–169. Springer International Publishing, Cham.

[Mankowska et al., 2014] Mankowska, D. S., Meisel, F., and Bierwirth, C. (2014). The home health care routing and scheduling problem with interdependent services. *Health Care Management Science*, 17(1):15–30.

[Martinez et al., 2018] Martinez, C., Espinouse, M.-L., and Di Mascolo, M. (2018). Continuity of care in home services: a client-centered heuristic for the home health care routing and scheduling problem. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1045–1050.

[Martinez et al., 2019] Martinez, C., Espinouse, M.-L., and Mascolo, M. D. (2019). Re-planning in home healthcare: A decomposition approach to minimize idle time for workers while ensuring continuity of care. *IFAC-PapersOnLine*, 52(13):654–659. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

[Mascolo et al., 2018] Mascolo, M. D., Espinouse, M., and Haddadene, S. A. (2018). Taking patients' wishes into account for daily planning in the home health care contex. *IFAC-PapersOnLine*, 51(11):1010–1015. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[Moussavi et al., 2019] Moussavi, S., Mahdjoub, M., and Grunder, O. (2019). A matheuristic approach to the integration of worker assignment and vehicle routing problems: Application to home healthcare scheduling. *Expert Systems with Applications*, 125:317–332.

[Nadine Genet and Saltman, 2012] Nadine Genet, Wienke Boerma, M. K. A. H. and Saltman, R. (2012). *Home care across Europe. Current structure and future challenges. In: Observatory Studies Series 27.* NIVEL.

[Nasir and Dang, 2018] Nasir, J. A. and Dang, C. (2018). Solving a more flexible home health care scheduling and routing problem with joint patient and nursing staff selection. *Sustainability*, 10(1).

[Nickel et al., 2012] Nickel, S., Schröder, M., and Steeg, J. (2012). Mid-term and short-term planning support for home health care services. *European Journal of Operational Research*, 219(3):574–587. Feature Clusters.

[Ouertani et al., 2019] Ouertani, N., Nouaouri, I., Ben-Romdhane, H., Allaoui, H., and Krichen, S. (2019). A hypermutation genetic algorithm for the dynamic home health-care routing problem. In *2019 International Conference on Industrial Engineering and Systems Management (IESM)*, pages 1–6.

[Redjem and Marcon, 2016] Redjem, R. and Marcon, E. (2016). Operations management in the home care services: a heuristic for the caregivers' routing problem. *Flexible Services and Manufacturing Journal*, 28(1):280–303.

[Resende and Ribeiro, 2005] Resende, M. and Ribeiro, C. (2005). Grasp: Greedy randomized adaptive search procedures. *Search Methodologies*.

[Rest and Hirsch, 2015] Rest, K.-D. and Hirsch, P. (2015). Supporting urban home health care in daily business and times of disasters. *IFAC-PapersOnLine*, 48(3):686–691. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[Rest and Hirsch, 2016] Rest, K.-D. and Hirsch, P. (2016). Daily scheduling of home health care services using time-dependent public transport. *Flexible Services and Manufacturing Journal*, 28(3):495–525.

[Riazi et al., 2017] Riazi, S., Wigström, O., Bengtsson, K., and Lennartson, B. (2017). Decomposition and distributed algorithms for home healthcare routing and scheduling problem. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–7.

[Riazi et al., 2019] Riazi, S., Wigström, O., Bengtsson, K., and Lennartson, B. (2019). A column generation-based gossip algorithm for home healthcare routing and scheduling problems. *IEEE Transactions on Automation Science and Engineering*, 16(1):127–137.

[Shahnejat-Bushehri et al., 2019] Shahnejat-Bushehri, S., Tavakkoli-Moghaddam, R., Momen, S., Ghasemkhani, A., and Tavakkoli-Moghaddam, H. (2019). Home health care routing and scheduling problem considering temporal dependencies and perishability with simultaneous pickup and delivery. *IFAC-PapersOnLine*, 52(13):118–123. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

[Shi et al., 2019] Shi, Y., Boudouh, T., and Grunder, O. (2019). A robust optimization for a home health care routing and scheduling problem with consideration of uncertain travel and service times. *"Transportation Research Part E: Logistics and Transportation Review"*, 128:52–95.

[Spinelli and Pellino, 2020] Spinelli, A. and Pellino, G. (2020). Covid-19 pandemic: perspectives on an unfolding crisis. *Journal of British Surgery*, 107(7):785–787.

[World Health Organization,] World Health Organization, year=2020, t.

[Xiao et al., 2018] Xiao, L., Dridi, M., and El Hassani, A. H. (2018). Mathematical model for the home health care scheduling and routing problem with flexible lunch break requirements. *IFAC-PapersOnLine*, 51(11):334–339. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[Xie and Wang, 2017] Xie, Z. and Wang, C. (2017). A periodic repair algorithm for dynamic scheduling in home health care using agent-based model. In *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 245–250.

[Yalçındağ and Matta, 2017] Yalçındağ, S. and Matta, A. (2017). A decomposition approach for the home health care problem with time windows. In Cappanera, P., Li, J., Matta, A., Sahin, E., Vandaele, N. J., and Visintin, F., editors, *Health Care Systems Engineering*, pages 221–232. Springer International Publishing, Cham.

[Yalçındağ et al., 2016] Yalçındağ, S., Cappanera, P., Grazia Scutellà, M., Şahin, E., and Matta, A. (2016). Pattern-based decompositions for human resource planning in home health care services. *Computers Operations Research*, 73:12–26.

[Yuan and Jiang, 2017] Yuan, B. and Jiang, Z. (2017). Disruption management for the real-time home caregiver scheduling and routing problem. *Sustainability*, 9(12).

[Yuan et al., 2015] Yuan, B., Liu, R., and Jiang, Z. (2015). A branch-and-price algorithm for the home health care scheduling and routing problem with stochastic service times and skill requirements. *International Journal of Production Research*, 53(24):7450–7464.

[Yuan et al., 2018] Yuan, B., Liu, R., and Jiang, Z. (2018). Daily scheduling of caregivers with stochastic times. *International Journal of Production Research*, 56(9):3245–3261.

Appendix A

METHODOLOGY

Algorithm 5: Lexicographic Goal Programming - VNS-2 Heuristic - Single HSP Case

Input : D : Distance matrix, P : Patients, $P^{'}$: The highest-prioritized patients, E : The beginning time of service providers, W : The ending time of service providers, T : The set of time blocks, $Route$: The initial route from the GRASP heuristic, $ArrivalTimes$: The initial arrival times from the GRASP heuristic

Output: $Route$ and $ArrivalTimes$ arrays

$bestRoute \leftarrow Route$;
$bestArr \leftarrow ArrivalTimes$;
$iter$, $nonImprovingIter \leftarrow 0$;
while $iter < iteratonLimit^1 \wedge nonImprovingIter < iteratonLimit^2$ **do**
    $tempRoute \leftarrow bestRoute$;
    $tempArr \leftarrow bestArr$;
    $lastRoute$, $lastArr \leftarrow twoExchange(lastRoute, lastArr)$;
    $lastRoute$, $lastArr \leftarrow oneDrop(bestRoute, bestArr)$;
    $lastRoute$, $lastArr \leftarrow oneAdd(bestRoute, bestArr)$;
    $tempRoute \leftarrow bestRoute$;
    if $isBest(lastRoute, lastArr, bestRoute, bestArr)$ **then**
        $bestRoute \leftarrow lastRoute$;
        $bestArr \leftarrow lastArr$;
        $nonImprovingIter \leftarrow 0$;
    endif
    $lastRoute$, $lastArr \leftarrow twoOpts(lastRoute, lastArr, D, E, W)$;
    if $isBest(lastRoute, lastArr, bestRoute, bestArr)$ **then**
        $bestRoute \leftarrow lastRoute$;
        $bestArr \leftarrow lastArr$;
        $nonImprovingIter \leftarrow 0$;
    endif
    $nonImprovingIter \leftarrow nonImprovingIter + 1$;
    $iter \leftarrow iter + 1$;
end
$bestPriority$, $estTotalTime \leftarrow getObjectiveValues(bestRoute, bestArr)$;
return $bestRoute$, $bestArrivalTime$, $bestPriority$, $bestTotalTime$

Appendix B

COMPUTATIONAL STUDY

Table B.1: Simulation results: Dynamic Problem Detailed Comparisons by Sample Size.

		Goal 1					Goal 2				
Size	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
20	Avg	114.49	113.10	108.68	114.39	108.47	23.53	40.94	37.29	34.98	41.76
	Min	98.00	53.00	57.00	98.00	92.00	7.43	14.85	11.11	11.65	16.92
	Max	133.00	130.00	131.00	132.00	127.00	49.58	77.97	63.03	64.44	74.92
	Std Dev	9.09	10.85	12.74	9.16	8.91	9.78	13.90	11.26	13.18	14.35
40	Avg	123.56	122.12	119.57	122.55	116.85	37.04	45.22	41.22	38.60	45.00
	Min	111.00	94.00	94.00	107.00	99.00	12.84	22.62	15.41	13.66	20.07
	Max	139.00	136.00	137.00	138.00	136.00	62.43	72.08	71.26	71.16	78.49
	Std Dev	8.25	8.07	8.83	8.50	7.59	13.47	12.47	13.61	13.94	15.11
60	Avg	128.98	127.92	123.40	127.65	121.08	36.16	43.53	40.82	37.78	43.83
	Min	114.00	114.00	89.00	113.00	107.00	12.34	15.75	12.08	13.10	18.27
	Max	143.00	142.00	139.00	141.00	137.00	68.73	75.71	77.93	73.25	80.54
	Std Dev	8.19	7.43	9.92	8.32	7.85	14.52	14.19	15.12	14.66	15.95

		Goal 3					Goal 4				
Size	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
20	Avg	3.97	4.18	3.08	5.38	3.67	39.97	39.98	38.58	41.38	39.67
	Min	0.70	0.00	0.00	2.83	0.98	36.70	27.53	31.55	38.83	36.98
	Max	7.01	6.66	6.08	7.91	6.36	43.01	42.66	42.08	43.91	42.36
	Std Dev	1.41	1.64	1.64	1.56	1.55	1.41	2.24	2.34	1.56	1.55
40	Avg	5.58	4.26	3.37	5.51	3.79	41.58	40.15	39.14	41.51	39.79
	Min	3.26	0.00	0.38	2.55	0.74	39.26	32.27	34.78	38.55	36.74
	Max	8.00	6.55	6.55	8.00	6.59	44.00	42.55	42.55	44.00	42.59
	Std Dev	1.56	1.49	1.47	1.58	1.58	1.56	1.72	1.66	1.58	1.58
60	Avg	5.64	4.30	3.37	5.65	3.90	41.64	40.23	38.90	41.65	39.90
	Min	114.00	3.30	1.24	0.00	2.88	1.05	39.30	37.24	32.40	38.88
	Max	8.00	6.47	6.55	8.00	7.48	44.00	42.47	42.55	44.00	43.48
	Std Dev	1.58	1.41	1.32	1.58	1.64	1.58	1.49	1.85	1.58	1.64

Table B.2: Simulation results: Dynamic Problem Detailed Comparisons by Sample Dynamism Level.

		Goal 1					Goal 2				
Dynamism Level	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
Low	Avg	112.72	112.64	109.70	110.96	107.68	21.36	29.49	27.29	22.53	27.80
	Min	98.00	98.00	92.00	98.00	92.00	11.34	14.85	11.11	11.65	16.92
	Max	125.00	125.00	121.00	121.00	121.00	33.68	48.08	48.78	32.72	63.21
	Std Dev	6.82	6.76	6.31	6.15	7.60	4.85	6.66	7.39	5.31	6.43
Mid	Avg	122.82	121.79	117.24	121.58	116.12	35.75	43.09	40.16	36.61	43.23
	Min	110.00	53.00	78.00	110.00	96.00	22.40	24.92	19.89	22.07	24.73
	Max	133.00	133.00	131.00	132.00	132.00	48.81	59.77	67.27	51.23	60.46
	Std Dev	6.40	9.58	9.56	5.90	7.68	6.18	8.34	9.33	6.95	7.80
High	Avg	131.91	128.62	124.14	131.52	122.06	39.90	56.99	52.48	52.66	60.09
	Min	105.00	87.00	57.00	121.00	97.00	7.43	33.03	29.29	34.80	34.80
	Max	143.00	142.00	139.00	141.00	137.00	68.73	77.97	77.93	73.25	80.54
	Std Dev	7.57	9.34	15.10	5.56	7.69	19.14	8.63	9.14	7.61	8.03

		Goal 3					Goal 4				
Dynamism Level	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
Low	Avg	3.73	2.71	2.21	3.64	1.99	39.73	38.54	37.81	39.64	37.99
	Min	2.82	0.93	0.00	2.55	0.74	38.82	36.93	33.85	38.55	36.74
	Max	4.43	3.75	3.31	4.48	6.40	40.43	39.75	39.25	40.48	42.40
	Std Dev	0.34	0.47	0.58	0.38	0.70	0.34	0.50	0.81	0.38	0.70
Mid	Avg	5.66	4.45	3.29	5.58	3.91	41.66	40.34	38.89	41.58	39.91
	Min	4.94	0.00	0.00	4.73	2.69	40.94	27.53	31.55	40.73	38.69
	Max	6.38	6.15	4.80	6.63	5.67	42.38	42.15	40.80	42.63	41.67
	Std Dev	0.33	0.82	1.08	0.45	0.65	0.33	1.53	1.75	0.45	0.65
High	Avg	5.83	5.56	4.37	7.36	5.52	41.83	41.47	39.99	43.36	41.52
	Min	0.70	0.00	0.00	5.93	4.19	36.70	32.20	32.34	41.93	40.19
	Max	8.00	6.66	6.55	8.00	7.48	44.00	42.66	42.55	44.00	43.48
	Std Dev	2.46	1.41	1.72	0.47	0.61	2.46	1.82	2.44	0.47	0.61

Table B.3: The Statistical Analysis of the Methods Used in Dynamic Problem through Performance Metrics.

			Total Run Time					# Post.					# HSP Changes				
Size	Dynamism Level	Stats.	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & HYBRID	MILP & MILP	MILP & G+V	G+V & MILP	G+V & G+V	G+V & & HYBRID
20	Low	Avg	73.67	61.14	83.36	71.13	48.61	0.00	0.00	0.00	0.13	0.93	1.17	2.20	1.53	2.27	1.67
		Min	1.12	6.07	1.85	15.26	2.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Max	180.98	152.45	180.80	135.46	105.31	0.00	0.00	0.00	2.00	2.00	4.00	8.00	4.00	8.00	5.00
		Std Dev	54.53	27.44	49.24	30.51	29.89	0.00	0.00	0.00	0.50	0.89	1.13	1.92	0.99	1.84	1.25
	Mid	Avg	100.85	74.57	97.51	91.63	58.12	0.00	0.48	0.00	0.97	1.39	2.57	5.70	2.77	3.63	3.13
		Min	4.40	36.63	3.84	35.06	3.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Max	340.48	138.07	238.38	158.49	150.46	0.00	14.00	0.00	11.00	4.00	7.00	15.00	7.00	8.00	9.00
		Std Dev	77.55	28.67	65.86	30.80	37.13	0.00	2.55	0.00	2.37	1.35	1.48	3.57	1.80	2.34	2.08
	High	Avg	207.32	132.24	193.84	134.02	123.37	0.47	0.80	0.04	2.57	2.14	1.40	11.30	4.20	7.90	4.50
		Min	33.74	60.74	28.48	42.43	22.24	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00
		Max	442.60	194.88	461.49	219.62	238.14	5.00	9.00	1.00	16.00	6.00	7.00	26.00	12.00	20.00	15.00
		Std Dev	119.31	34.57	110.89	45.92	56.60	1.41	1.68	0.19	4.48	1.38	1.72	5.75	2.56	5.15	3.38
40	Low	Avg	121.35	71.61	115.61	76.89	61.84	0.00	0.04	0.00	0.07	0.67	2.33	5.83	2.90	5.83	4.03
		Min	7.50	0.00	9.39	21.04	9.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
		Max	240.35	151.25	240.30	138.91	123.74	0.00	1.00	0.00	1.00	2.00	7.00	18.00	8.00	13.00	10.00
		Std Dev	58.70	30.15	55.34	30.44	27.15	0.00	0.19	0.00	0.25	0.79	1.53	4.42	1.74	3.62	2.52
	Mid	Avg	195.12	106.50	207.67	122.68	104.17	0.00	0.00	0.00	0.72	1.21	3.57	10.63	3.53	8.30	5.40
		Min	25.40	47.79	63.26	25.90	18.73	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00
		Max	319.50	220.52	334.03	232.50	183.23	0.00	0.00	0.00	7.00	4.00	8.00	33.00	11.00	17.00	24.00
		Std Dev	73.77	33.27	80.92	50.69	42.54	0.00	0.00	0.00	1.64	0.98	1.41	6.78	2.23	4.34	4.45
	High	Avg	293.90	133.90	246.69	145.28	142.28	0.12	1.08	0.04	0.74	2.31	5.30	15.83	5.20	14.30	7.43
		Min	33.69	58.78	0.00	64.87	47.94	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	2.00	2.00
		Max	480.84	200.62	454.14	306.54	260.20	1.00	10.00	1.00	7.00	7.00	12.00	36.00	14.00	38.00	16.00
		Std Dev	130.14	43.35	123.65	63.62	53.66	0.32	2.30	0.19	1.51	1.42	2.57	9.54	2.93	6.93	3.03
60	Low	Avg	97.50	49.75	93.34	46.86	40.59	0.00	0.04	0.00	0.50	0.76	2.50	8.53	2.63	5.27	3.73
		Min	7.14	11.61	8.43	10.27	11.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Max	197.15	114.31	191.18	120.03	84.72	0.00	1.00	0.00	4.00	2.00	9.00	20.00	17.00	15.00	20.00
		Std Dev	61.15	26.68	57.86	29.13	22.31	0.00	0.19	0.00	1.12	0.81	1.95	5.43	3.15	4.10	4.21
	Mid	Avg	235.78	93.31	207.72	88.35	86.81	0.00	0.07	0.00	0.48	1.19	3.93	14.97	5.37	12.97	7.33
		Min	32.07	32.78	10.41	20.86	15.20	0.00	0.00	0.00	0.00	0.00	1.00	5.00	0.00	0.00	0.00
		Max	360.56	154.15	360.45	183.91	140.48	0.00	1.00	0.00	3.00	4.00	7.00	34.00	12.00	40.00	21.00
		Std Dev	90.43	34.08	103.87	45.31	30.29	0.00	0.25	0.00	0.90	1.07	1.69	7.13	3.19	8.93	4.66
	High	Avg	324.35	126.11	294.78	100.24	134.04	0.15	0.70	0.08	0.96	2.08	6.17	23.57	7.37	19.27	7.80
		Min	119.09	34.63	28.30	35.10	28.30	0.00	0.00	0.00	0.00	0.00	1.00	7.00	0.00	0.00	0.00
		Max	480.72	208.47	480.79	203.05	279.02	1.00	6.00	1.00	6.00	4.00	12.00	42.00	24.00	43.00	18.00
		Std Dev	107.75	44.34	114.15	44.15	55.50	0.36	1.35	0.27	1.48	1.29	2.61	9.00	5.46	10.01	4.73

Table B.4: Simulation results: Dynamic Problem Detailed Comparisons of Performance Metrics by Sample Sizes.

		Total Run Time					# **Post.**				
Size	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
20	Avg	127.28	89.81	121.98	98.20	76.59	0.16	0.43	0.01	1.20	1.48
	Min	1.12	6.07	1.85	15.26	2.47	0.00	0.00	0.00	0.00	0.00
	Max	442.60	194.88	461.49	219.62	238.14	5.00	14.00	1.00	16.00	6.00
	Std Dev	105.20	43.45	91.69	44.68	54.12	0.84	1.80	0.11	3.07	1.32
40	Avg	198.14	102.98	188.43	113.80	102.28	0.04	0.34	0.01	0.50	1.39
	Min	7.50	0.00	0.00	21.04	9.37	0.00	0.00	0.00	0.00	0.00
	Max	480.84	220.52	454.14	306.54	260.20	1.00	10.00	1.00	7.00	7.00
	Std Dev	113.58	43.56	105.83	57.21	53.80	0.18	1.36	0.11	1.32	1.29
60	Avg	215.11	90.63	199.82	79.19	86.51	0.05	0.27	0.03	0.65	1.33
	Min	7.14	11.61	8.43	10.27	11.28	0.00	0.00	0.00	0.00	0.00
	Max	480.72	208.47	480.79	203.05	279.02	1.00	6.00	1.00	6.00	4.00
	Std Dev	128.51	47.50	126.56	46.34	53.81	0.21	0.86	0.16	1.22	1.20

		# HSP Changes				
Size	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
20	Avg	1.71	6.40	2.83	4.60	3.10
	Min	0.00	0.00	0.00	0.00	0.00
	Max	7.00	26.00	12.00	20.00	15.0
	Std Dev	1.59	5.53	2.19	4.19	2.67
40	Avg	3.73	10.77	3.88	9.48	5.62
	Min	0.00	0.00	0.00	0.00	0.00
	Max	12.00	36.00	14.00	38.00	24.00
	Std Dev	2.26	8.30	2.54	6.27	3.71
60	Avg	4.20	15.69	5.12	12.50	6.29
	Min	0.00	0.00	0.00	0.00	0.00
	Max	12.00	42.00	24.00	43.00	21.00
	Std Dev	2.60	9.58	4.51	9.92	4.89

Table B.5: Simulation results: Dynamic Problem Detailed Comparisons of Performance Metrics by Sample Dynamism Levels.

		Total Run Time					# Post.				
Size	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
Low	Avg	97.50	60.84	97.52	66.25	50.92	0.00	0.02	0.00	0.21	0.79
	Min	1.12	0.00	1.85	10.27	2.47	0.00	0.00	0.00	0.00	0.00
	Max	240.35	152.45	240.30	138.91	123.74	0.00	1.00	0.00	4.00	2.00
	Std Dev	61.36	29.49	55.73	32.59	28.24	0.00	0.15	0.00	0.71	0.84
Mid	Avg	175.92	91.65	167.75	101.49	82.94	0.00	0.18	0.00	0.73	1.27
	Min	4.40	32.78	3.84	20.86	3.84	0.00	0.00	0.00	0.00	0.00
	Max	360.56	220.52	360.45	232.50	183.23	0.00	14.00	0.00	11.00	4.00
	Std Dev	98.46	34.68	98.85	45.72	41.81	0.00	1.48	0.00	1.78	1.15
High	Avg	272.25	130.56	244.49	126.93	133.18	0.26	0.85	0.05	1.44	2.18
	Min	33.69	34.63	0.00	35.10	22.24	0.00	0.00	0.00	0.00	0.00
	Max	480.84	208.47	480.79	306.54	279.02	5.00	10.00	1.00	16.00	7.00
	Std Dev	129.51	40.98	123.43	55.33	55.83	0.91	1.79	0.22	3.01	1.37

		# HSP Changes				
Size	**Stats.**	**MILP & MILP**	**MILP & G+V**	**G+V & MILP**	**G+V & G+V**	**G+V & HYBRID**
Low	Avg	2.00	5.52	2.36	4.46	3.14
	Min	0.00	0.00	0.00	0.00	0.00
	Max	9.00	20.00	17.00	15.00	20.00
	Std Dev	1.68	4.93	2.23	3.68	3.11
Mid	Avg	3.36	10.43	3.89	8.30	5.29
	Min	0.00	0.00	0.00	0.00	0.00
	Max	8.00	34.00	12.00	40.00	24.00
	Std Dev	1.64	7.13	2.71	7.02	4.27
High	Avg	4.29	16.90	5.59	13.82	6.58
	Min	0.00	0.00	0.00	0.00	0.00
	Max	12.00	42.00	24.00	43.00	18.00
	Std Dev	3.12	9.70	4.09	8.94	4.06

Table B.6: The Comparison of Cooperation and Non-Cooperation by Samples' Statistical Analysis of Goals, Run Time, and Postponements.

Sample Dynamism Level	Stats.	Non-Cooperation					Cooperation					
		Total Priority	**Arrival Time Changes**	**Overtime Overtime**	**Total Time**	**# Post.**	**Total Priority**	**Arrival Time Changes**	**Total Overtime**	**Total Time**	**# Post.**	**HSP Changes**
Low	Avg	104.50	9.04	0.92	36.22	1.25	106.75	22.72	1.12	36.84	0.25	2.50
	Min	100.00	4.20	0.45	35.25	0.00	100.00	10.54	0.00	34.89	0.00	1.00
	Max	109.00	11.50	1.16	37.16	2.00	112.00	35.08	2.24	38.24	1.00	4.00
	Std Dev	3.64	2.84	0.28	0.89	0.83	4.66	9.18	0.87	1.27	0.43	1.50
Mid	Avg	106.67	12.96	1.91	37.82	1.00	111.83	30.51	1.90	37.40	0.67	0.67
	Min	97.00	7.64	0.89	36.34	0.00	102.00	10.88	0.18	35.08	0.00	0.00
	Max	114.00	15.80	2.43	38.43	3.00	123.00	47.09	4.07	39.67	1.00	1.00
	Std Dev	5.68	2.89	0.60	0.77	1.15	7.06	12.63	1.42	1.57	0.47	0.47
High	Avg	112.44	13.15	1.97	37.92	2.22	117.67	41.30	3.31	38.33	0.44	1.00
	Min	101.00	7.07	0.87	36.63	0.00	101.00	11.63	0.00	34.96	0.00	0.00
	Max	123.00	15.76	3.14	39.14	4.00	129.00	62.22	4.07	39.97	2.00	3.00
	Std Dev	7.15	2.99	0.73	0.81	1.23	9.24	16.01	1.94	1.66	0.68	1.05

Table B.7: The Results of the 3-HSPs' Cooperation with New Emergent Patients.

Patient	Sample	3-HSPs' Cooperation with New Emergent Patients					
		Goal 1	Goal 2	Goal 3	Goal 4	# Post.	# HSP Changes
1	40-Low-7	136	12.23	2.02	48.78	0	5
2		141	4.96	2.01	49,71	0	1
3		146	5.62	2.36	50.36	0	2
4		151	7.74	2.98	50.98	0	4
5		156	9.40	3.94	51.94	0	3
6		161	10.21	4.84	52.84	0	1
7	40-Mid-0	135	10.89	0.55	48.38	0	3
8		140	5.09	1.26	49.26	0	2
9		145	6.52	2.10	50.10	0	2
10		144	7.51	2.35	50.35	2	0
11		149	11.56	3.09	51.09	2	0
12		150	15.32	4.74	52.74	3	1
13		155	13.50	5.52	53.52	3	3
14		160	15.97	6.39	54.39	3	0
15		165	18.54	7.26	55.26	3	0
16	40-High-8	139	31.79	2.98	50.93	0	6
17		144	7.22	3.81	51.81	0	6
18		149	11.26	4.76	52.76	0	9
19		154	8.05	5.22	53.22	0	6
20		159	10.65	5.87	53.87	0	2
21		164	10.23	6.68	54.68	0	1
22		169	10.40	7.41	55.41	0	0
23		174	10.96	8.28	56.28	0	1
24		179	12.73	9.09	57.09	0	2
25		184	13.57	9.85	57.85	0	4
26		189	15.41	10.83	58.83	0	1
27		190	16.56	11.17	59.17	1	1

Table B.8: The Summary of the Average Performance Metrics of 3-HSPs' Cooperation Instances.

Sample	3-HSPs' Cooperation with New Emergent Patients					
	Goal 1	**Goal 2**	**Goal 3**	**Goal 4**	**# Post.**	**# HSP Changes**
40-Low-7	148.50	8.36	3.025	50.98	0	2.67
40-Mid-0	149.23	11.66	3.70	51.68	1.78	1.23
40-High-8	166.17	13.24	7.16	55.16	0.08	3.25