Heuristic approaches for the Home Health Care staff scheduling problem

Arshia Ghavampour
Supervisor: Jonathan Patrick

Thesis Proposal to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of science Systems science engineering

Faculty of Engineering
University of Ottawa

Abstract

The increasing demand for community-based healthcare services requires efficient scheduling of caregivers to ensure high-quality, consistent care for clients while minimizing costs. This thesis addresses the challenge of creating optimal schedules for Personal Support Workers (PSWs) in supportive housing complexes, focusing on maximizing consistency for clients and reducing reliance on external staffing agencies. The problem is formulated as a Mixed Integer Linear Programming (MILP) model, incorporating specific objectives and constraints that assign PSWs to patient requests. Exact methods such as Branch and Bound(B&B) and Branch and Cut(B&C) were initially explored but, due to the complexity of the problem, these proved incapable of providing timely solutions. To overcome this, heuristic approaches, including Logic-Based Benders Decomposition and Lagrangian Relaxation were employed, where the solution process was stopped prematurely, effectively truncating the solution. As a case study, the proposed model is applied to Nucleus Independent Living (NIL), an organization based in Oakville, which aims to improve its scheduling practices for at-home and supportive housing services. The computational results demonstrate that Benders Decomposition outperforms Lagrangian Relaxation, achieving a solution within 0.2% of the best result obtained by the commercial solver GUROBI.

Table of Contents

List of Figures and Tables

1. Introduction

As the senior population grows, the consumption of health services by older individuals has likewise risen, particularly in the setting of home health care, HHC. HHC encompasses a range of medical services provided directly to patients within their home. Its primary objective is to aid individuals with medical needs, disabilities, or chronic health conditions who may encounter obstacles in accessing medical assistance outside their homes. These services comprise medical treatment, nursing care, therapy sessions, and assistance with daily activities, all tailored to meet the specific needs of each patient. HHC plays an integral role in enhancing the quality of life for individuals requiring ongoing medical attention and preferring the familiar and comforting environment of their own home. It embraces a personalized and patient-centric approach, enabling individuals to receive treatment while preserving their independence, dignity, and emotional well-being. By receiving care in their home, patients can mitigate the necessity for hospitalizations, decrease the risk of infections, and maintain a sense of normalcy in their day-to-day routines.

Home healthcare services typically include nursing or support care. Registered nurses offer a broad spectrum of medical services such as wound care, medication management, intravenous therapy, and disease education. Personal support workers (PSW) provide assistance with bathing, grooming, dressing, toileting, and other hygiene tasks, allowing patients to maintain their independence and dignity. Additionally, medical social workers offer counseling, emotional support, and guidance on accessing community resources to tackle social, emotional, and financial issues.

Home healthcare presents numerous benefits for patients. It enables them to receive tailored care within the comfort of their familiar surroundings, promoting enhanced outcomes and overall satisfaction. Patients can preserve their autonomy, participate in daily activities, and adhere to established routines. Moreover, home healthcare services typically offer a more economical alternative to institutional care settings such as hospitals or long-term care facilities, leading to significant cost savings for both patients and healthcare systems.

However, HHC providers face worker shortages and high labor costs. According to Scheffler and Arnold [1] Canada is anticipated to confront a shortfall of 117,600 nurses by 2030. Moreover, based on Ontario Health Coalition reports, the shortage of PSWs is just as significant. Consequently, the importance of optimizing nurse/PSW scheduling is set to rise. One effective approach is to use operations research (OR) methodologies, which provide systematic and data-driven tools to optimize the allocation of nurses or PSWs to patients—a challenge commonly referred to as the Nurse Rostering Problem (NRP). OR methods were chosen for their ability to model complex scheduling constraints, such as patient continuity of care, while ensuring cost-effectiveness and high-quality care. The problem will be modeled as a mixed-integer linear programming (MILP) problem, incorporating real-world constraints and objectives to produce implementable and efficient schedules.

2. Literature review

In the past few decades, researchers have extensively explored the field of nurse scheduling and rostering. This literature review organizes the discussion into three main sections: (1) studies addressing nurse scheduling and Home Health Care (HHC), (2) specific methods applied to these problems, and (3) general optimization techniques and their applicability to the challenges raised in this thesis.

2.1. Nurse Scheduling Formulations and Constraints

Review articles offer an overview of a field, identifying significant gaps and challenges that future research should address. Fikar and Hirsch [2] provide a review that describes the complex nature of Home Health Care (HHC) routing and scheduling, highlighting the necessity for effective solutions to meet the changing needs within this sector. Ernst et al. [3] offer a broader review of staff rostering models across different fields, including nurse rostering. As defined in Burke et al. [4], 'nurse rostering' usually applies to the short-term timetabling of nurses (with a typical time horizon of a few weeks). Burke et al. [4] compares various approaches aimed at solving the nurse rostering problem. They consider aspects such as problem definition, methodology, criteria for evaluation, and practical applications in real-world settings.

These reviews underscore that while advanced algorithms have been developed to address the complexities of nurse scheduling, challenges remain in achieving scalability and flexibility to adapt to real-time changes, such as sudden staff shortages or fluctuating patient demands. Future research should focus on techniques to enhance the adaptability and responsiveness of scheduling systems, as well as exploring heuristic approaches to bridge the gap between optimality and computational efficiency.

Haspeslagh [5] initiated the First International Nurse Rostering Competition (INRC-I) in 2010. This event prompted researchers to address the common practical constraints faced in nurse scheduling and encouraged collaborative efforts to develop creative solutions. Additionally, it offered a platform to evaluate competing approaches. Haspeslagh [5] provides a description of the typical constraints faced in nurse scheduling, dividing them into hard and soft constraints. However, their model does not include multi-periods or continuity of care – both significant challenges in nurse scheduling. Single-stage problems can not account for long-term considerations or continuity between periods.

Ceschia and Dang [6], who organized the Second International Nurse Rostering Competition (INRC-II), presented a multi-stage problem formulation. This formulation requires that solutions of different stages (i.e. weeks) be consistent with each other, unlike previous research that often treated the NRP as a single-stage problem. Mischek and Musliu [7] developed a concise MIP method for the INRC-II problem. They added extra constraints to deal with the uncertainty caused by multiple stages. They used CPLEX to solve their improved model and found that it was as good as the best solutions of the INRC-II competition.

Additionally, shift or visit time preferences are major issues that can have a significant impact on the quality of work or the reputation of a HHC organization. Böðvarsdóttir and Bagger [8] provided a MIP model with additional constraints to create balanced work routines for all nurses. The proposed model aims to align with the specific needs of hospital units, considering qualifications, preferences, and working hours. It emphasizes the optimization of nurse efficiency while meeting demand. Guo and Bard [9] developed a model that incorporates constraints on consecutive work or rest days for nurses, minimum intervals between shifts, and soft constraints related to nurse preferences for shifts.

2.2 Methods

While modeling a NRP can be done through various approaches, mixed integer linear programming (MILP) is the most popular though the size of the problem often requires heuristic methods. DenHartog and HanHoogeveena [11] presented a Mixed Integer Programming (MIP) framework designed to reconcile the competing demands of nurse availability, preferences, qualifications, and working hours. Utilizing a MIP solver implemented in Python 3, they aim to identify the optimal solution using techniques such as cutting planes.

When it comes to solving an optimization problem, it is important to find a solution in an acceptable amount of time - often requiring heuristic approaches. Gomory [12] introduced a pioneering algorithm to refine linear relaxations by iteratively adding cuts to eliminate fractional solutions. Boyd and Vandenberghe [13] expand on this, discussing cutting planes within the broader context of convex optimization, emphasizing their role in tightening feasible regions and improving solution quality for complex problems. Guo and Bard [9], divided the HHC problem into two stages, due to the complexity of simultaneously handling regular and overtime hours. In the first part, they employed a mathematical model to assign shifts to nurses without exceeding their regular hours, utilizing column generation to find solutions. In the second part, a heuristic method is employed to supplement the schedules generated in the first part with additional hours as needed. Burke and Curtois [14] used a branch and price technique to tackle the INRC-I scenarios. They introduced a general rostering model integrating many of the popular constraints through pattern matching. Their approach categorized the primary problem as a set covering one, while breaking it down into subproblems resembling the shortest path problem with resource constraints. To solve these subproblems, they utilized dynamic programming alongside heuristic principles.

Santos et al. [15] applied IP methods to the INRC-I problems. They introduced the concept of working and resting windows. Working windows indicate when nurses are scheduled to work, while resting windows outline guaranteed rest times, ensuring a balanced distribution of work and rest throughout the roster. Leveraging this alternating pattern, they introduced additional cuts to fortify the problem formulation. To expedite the process of finding near-optimal solutions, they employed a MIP heuristic. This approach achieved optimality for the majority of INRC-I instances and yielded superior solutions for other problems. Guessoum et al. [16] proposed a two-phase model for nurse rostering where the first phase involves using a generic variable fixing heuristic to create a second sub-problem. This sub-problem is then solved using a general-purpose MIP solver, resulting in an efficient optimization approach. Valouxis et al. [17] the winner of INRC-I, applied a two-stage approach to address the problem. They decomposed the problem into smaller components and addressed them individually using free integer programming software. Initially, they determined the days on which nurses would work, followed by assigning shifts for each day in the subsequent step. Additionally, they utilized heuristic techniques to improve solutions.

2.3 Generic Optimization Techniques and Their Applicability

Several general optimization techniques have been applied to nurse scheduling and HHC. Desaulniers et al. [18] provide a comprehensive overview, detailing the application of column generation across various industries, while Lübbecke and Desrosiers (2005) [19] delve into specific topics, highlighting its efficiency in dealing with massive sets of variables by dynamically generating columns during the optimization process.

Marshall L. Fisher [20] introduced a powerful technique that simplifies complex problems by relaxing difficult constraints using Lagrange multipliers. This method is particularly effective for large-scale integer programming, providing bounds and facilitating the use of heuristic methods. While not always yielding exact solutions, it has significantly influenced the development of advanced optimization techniques like cutting-plane methods and branch-and-bound algorithms.

By applying Lagrangian relaxation, A. M. Fathollahi-Fardard et al. [21] decompose the HHC scheduling problem into more manageable subproblems, effectively improving computational efficiency. The method yields strong bounds and feasible solutions, making it well-suited for large-scale instances of the HHC routing problem. This approach highlights the adaptability of Lagrangian relaxation in practical, real-world optimization scenarios.

J. F. Bender [22] introduced the Benders Decomposition (BD) algorithm, primarily designed to address problems with complicating variables. By temporarily fixing these variables, the resulting problem becomes easier to solve. Also known as variable partitioning and outer linearization, BD has become one of the most widely used exact algorithms, as it leverages the problem's structure and distributes the computational workload.

Although BD is mostly used in linear problems, J.N. Hooker and G. Ottosson [23] provide a Logic-based Bender's decomposition (LBBD) that presents a generalized approach to traditional Bender's decomposition, targeting combinatorial optimization problems. The method effectively integrates mixed-integer programming (MIP) and constraint programming (CP), utilizing their complementary strengths. Hooker demonstrates the applicability of LBBD in various domains, particularly in scheduling and planning problems. Curtiss Luong [24] used this method to create two innovative Benders' algorithms - a traditional Benders' algorithm and a combinatorial version - to effectively address the sector duration and optimization problem. They apply a pre-existing logic-based Benders' algorithm for tactical decision-making and assess how variations in input data impact different output metrics. The findings revealed that their two-phase Benders' decomposition offered computational benefits compared to a conventional branch-and-cut method, especially when tackling extremely large problems.

For a HHC setting, Bahman Naderi et al. [25] introduced the Branch and Check approach that first solves a deterministic problem and then identifies a robust solution. The master problem (MP) determines which caregiver will serve which patients on each day, while the subproblem (SP) schedules the visits. Simple monotone optimality cuts are generated for each caregiver. If the SP takes too long to solve, it is solved approximately, retaining its upper bound, and the cut is saved for later. Once Branch and Check concludes, the search process branches based on whether or not to enforce each saved cut, solving the SP to optimality. Cuts are removed when they are no longer effective given the current upper bound.

The field of nurse scheduling and rostering is complex and requires balancing numerous constraints and preferences, such as shift patterns, coverage requirements, and staff preferences. Unique to this problem is the need to accommodate various types of Personal Support Workers (PSWs) – including Full-Time (FT), Part-Time (PT), Permanent Part-Time (PPT), and Agency staff – each with different availability, salaries, and contractual obligations. This diversity adds complexity to the scheduling process, as it must account for different working hours, preferences, and varying levels of commitment. Researchers have made significant strides in addressing these challenges. Our goal is to model the real-world problem faced by a HHC provider and develop a practical approach that can efficiently find a solution within an acceptable timeframe.

3. Problem statement and Research Questions

In this research, the NRP problem is defined by challenges faced by Nucleus Independent Living (NIL), an organization situated in Oakville that offers in-home and supportive housing services to clients by deploying Personal Support Workers (PSWs). Amongst other clients, NIL services two apartment buildings in Toronto dedicated to supportive housing. The Supportive Housing program has the capacity to accommodate 28 clients. NIL assigns PSWs daily to cover time-sensitive service requests from patients according to their ability and availability. Services are available around the clock and are categorized into three levels of care. The first level comprises tasks such as homemaking, meal preparation, and garbage collection. The second level includes bathing assistance and medication reminders. The final level encompasses delegated acts, such as catheter insertion. Each patient requires a different level and frequency of service. The purpose of this research is to provide an efficient schedule for PSWs in supportive housing complexes in order to minimize a weighted sum of the number of different PSWs who visit a patient, the time difference between the planned time of visit and preferred time of visit, and the cost of using non-Nucleus PSWs.

The model utilizes data from patients within the complex, taking into account several key factors: time window limits, service durations, and union regulations. Additionally, the scheduling must incorporate the different types of Personal Support Workers (PSWs) available - Full-Time (FT), Part-Time (PT), Permanent Part-Time (PPT), and Agency staff, each with unique availability, costs, and contractual conditions. Patients may have multiple service requests throughout the day, and some of these requests may require the simultaneous presence of more than one PSW.

While routing is a significant challenge in many HHC settings, it is not a primary concern in this supportive housing context due to the proximity of all patients, making travel times negligible. Instead, the focus is on optimizing the assignment of PSWs to ensure service continuity, and adherence to both patient preferences and regulatory constraints. These factors collectively pose significant challenges to the management and scheduling of healthcare services at NIL.

3.1 Research question

Our research will focus on a series of connected questions:

1. What is the best method to model the HHC scheduling problem encountered by NIL?
2. How can we solve this model?

Exact methods are of course preferred. However, due to the size of the problem, a heuristic approach can help to find an acceptable solution.

3. How can we ensure the approach finds an acceptable solution by comparing different heuristic approaches in reasonable amount of time?

A common approach is to compare the proposed solution to the exact method in a smaller setting and measure the optimality gap between the heuristic solution and the exact one

4. Mathematical model:

We have chosen to model the Home Healthcare (HHC) problem using a Mixed Integer Programming (MIP) approach due to its ability to handle complex combinatorial optimization problems and provide exact solutions. Unlike simulation, which may be more suitable for analyzing the impact of random variables over time, MIP allows for precise optimization of resource allocation, taking into account multiple objectives and constraints simultaneously. This level of precision is crucial to developing a reliable and efficient scheduling solution that maximizes the use of available resources while minimizing costs and improving patient care.

Moreover, we have opted for a deterministic model despite the presence of stochastic components, such as varying patient demand, uncertain service durations, and staff availability. This decision is based on the need for a practical and computationally feasible approach that can provide actionable scheduling plans within a reasonable timeframe. While stochastic models could capture these uncertainties more accurately, they often require significantly more computational effort and data, which may not be readily available. Instead, our deterministic model provides an implementable, and feasible solution that is, to a great extent, optimal for real-world use cases at NIL.

4.1 Indices:

Let i=1,2,…,I represent the set of Personal Support Workers (PSWs). The PSWs are categorized into different types:

- i=1,…, I_1 represent full-time PSWs,
- i=I_1+1,…, I_2 represent permanent part-time PSWs,
- i=I_2+1,…, I represent part-time PSWs.
- and more than that represent Agencies.

j= (1,2,…J) Represent the number of patients (clients)

r represents each 15-minute time slot in a given day (see Table 1 below), r ∈ (1,2…,96)

Table 1: Time slot in a day

r	time	r	time	r	time	r	time	r	time	r	time	r	time	r	time	r	time	r	time	r	time
1	0:00	10	2:15	19	4:30	28	6:45	37	9:00	46	11:15	55	13:30	64	15:45	73	18:00	82	20:15	91	22:30
2	0:15	11	2:30	20	4:45	29	7:00	38	9:15	47	11:30	56	13:45	65	16:00	74	18:15	83	20:30	92	22:45
3	0:30	12	2:45	21	5:00	30	7:15	39	9:30	48	11:45	57	14:00	66	16:15	75	18:30	84	20:45	93	23:00
4	0:45	13	3:00	22	5:15	31	7:30	40	9:45	49	12:00	58	14:15	67	16:30	76	18:45	85	21:00	94	23:15
5	1:00	14	3:15	23	5:30	32	7:45	41	10:00	50	12:15	59	14:30	68	16:45	77	19:00	86	21:15	95	23:30
6	1:15	15	3:30	24	5:45	33	8:00	42	10:15	51	12:30	60	14:45	69	17:00	78	19:15	87	21:30	96	23:45
7	1:30	16	3:45	25	6:00	34	8:15	43	10:30	52	12:45	61	15:00	70	17:15	79	19:30	88	21:45		
8	1:45	17	4:00	26	6:15	35	8:30	44	10:45	53	13:00	62	15:15	71	17:30	80	19:45	89	22:00		
9	2:00	18	4:15	27	6:30	36	8:45	45	11:00	54	13:15	63	15:30	72	17:45	81	20:00	90	22:15		

t represents the start time of a shift measured in number of 15-minute intervals and restricted based on current practice. We limit our analysis to the following available shifts:

Table 2: Shifts during a day

Shift	Time interval
Shift 1	from 12 am to 8 am (8h)
Shift 2	from 7 am to 1 pm (6h)
Shift 3	from 7 am to 3 pm (8h)
Shift 4	from 3 pm to 11 pm (8h)
Shift 5	from 4 pm to 12 am (8h)

d represents the days of the scheduling interval, d ∈ [1, 2, 3, ….., 14]

s represents the number of the service request, s ∈ (1, 2, …..)

4.2 Parameters & Sets:

ST: Set of service types [1, 2, 3, 4] (1: 15-minute service, 2: 30-minute, 3 :45-minute, 4: 60-minute)

S_d: Set of service requests on day d

E_j: Set of service requests for patient j

k_s: Preferred time of visit for client j for service request s

LB_s: Lower bound for the visit start time for service request s

UB_s: Upper bound for the visit start time for service request s

ST_s: Service Type for service request s

U_{ij}: If PSW 'i' is able to serve patient j, U_{ij}=1; otherwise, U_{ij}=0

C_i: Cost of PSW i per hour

H_i: Total working hours for PSW i over the 2-week period.

G_i: Extra allowed working hours for PSW i over the 2-week period

M: A large number (at least equal to the maximum possible value of the variable it bounds).

4.3 Decision variables:

Z_{ij}: If PSW i is assigned to do a service request for client j, then $Z_{ij} = 1$, else, $Z_{ij} = 0$

X_{idt}: If PSW 'i' starts a shift at time 't' on day 'd', then $X_{idt} = 1$, else $X_{idt} = 0$

Y_{irs}: If PSW 'i' starts service request 's' at time 'r', then $Y_{irs} = 1$, else $Y_{irs} = 0$

B_{id}: Number of working hours for PSW i on day d

W_s: Gap between scheduled time for service request s and the patient's preferred time

L_{ird}: If PSW 'i' starts his/her break at time 'r' on day 'd', then $L_{ird} = 1$, else $L_{ird} = 0$

4.4 Objectives:

Our setting involves multiple objectives – 1) minimize the gap between the scheduled visit time and preferred one, 2) minimize the number of assigned Agency PSWs and 3) minimize the number of PSWs assigned to service requests for each client (continuity of care). We utilize the weighted sum approach to address the multi-objective nature of the problem. This method is a widely used due to its ease of implementation. By assigning weights to different objectives, it allows decision-makers to prioritize goals based on their relative importance. This method is computationally efficient and can be easily integrated into existing optimization solvers. It is also appropriate when the relative importance of the objectives can be easily established.

In contrast, Pareto-based approaches [26], such as the Pareto front search, provide a trade-off analysis but can be computationally expensive, especially for problems with many objectives. They also do not provide the user with a single solution (as necessitated by NIL) but rather leave the client to choose between a set of possible solutions. ε-constraint methods convert a multi-objective problem into a single-objective by constraining all but one objective, which can provide more flexibility but requires iterative adjustments to constraints and a judgement call as to the appropriate bounds on the constraints. Finally, goal programming seeks to minimize the deviation from predefined target values for each objective but depends heavily on the choice of targets. While these alternative methods offer unique advantages, the weighted sum approach is particularly effective when a clear understanding of the relative importance of objectives exists and when the focus is on obtaining a single, preferred solution quickly. [27]

4.5 Objective function:

We therefore formulate the following normalized, weighted multi-objective function:

$$\textbf{Min}\ \left(\mathbf{w_1}\left(\frac{\sum_{s\in S} W_s}{\mathbf{W_{max}}}\right)+\mathbf{w_2}\left(\frac{\sum_{i\in(FT,PPT,PT)}\sum_{d\in D} C_i*\mathbf{B_{id}}+\sum_{i\in(Agency)}\sum_{d\in D} C_i*\mathbf{B_{id}}}{\mathbf{B_{max}}}\right)+\mathbf{w_3}\left(\frac{\sum_{i\in I}\sum_{j\in J}\mathbf{Z_{ij}}}{\mathbf{Z_{max}}}\right)\right) \quad \textbf{(1)}$$

where w_1, w_2, w_3 represent the weights of each objective and W_{max}, B_{max}, and Z_{max} represent the maximum value that each of the objectives can achieve.

W_{max} can be calculated by summing the deviations between the latest and earliest possible visit times for all service requests in S.

B_{max} is equal to the sum of the maximum number of work hours for each type of worker per day. Finally, Z_{max} is equal to the sum of the maximum number of interactions for each PSW-client pair.

4.6 Constraints:

$\sum_{i\epsilon I} X_{idt} \geq 2 \qquad \forall\ d \in D, \forall\ t \in T \qquad (2)$

$\sum_{i\in I} Z_{ij} \geq 1 \qquad \forall j \epsilon J \qquad (3)$

$\sum_{s\epsilon E_j} Y_{irs} \leq M*Z_{ij} \qquad \forall i\epsilon I, \forall j\epsilon J, \forall r \in R \qquad (4)$

$\sum_{r\in R} Y_{irs} \leq U_{ij}$ ∀iϵI, ∀jϵJ (5)

$\sum_{i\epsilon I}\sum_{r \in R} Y_{irs}=1$ ∀sЄS (6)

$\sum_{r=1}^{t-1} L_{ird} \leq M(1-X_{idt})$ ∀ i ϵ I, ∀ d Є D, ∀ t ϵ T (7)

$\sum_{r\in R}(r+1)L_{ird} \leq \sum_{t \in T}(t * X_{idt}) + 4*B_{id}$ ∀ i ϵ I, ∀ d Є D (8)

$\sum_{s \in Sd}\sum_{r''=r}^{r'+1} Y_{ir''s} \leq M*(1-L_{ir'd})$ ∀ i ϵ I, ∀ r' Є R, ∀ s Є S, ∀ d Є D (9)

$B_{id}-4 \leq 5*\sum_{r\in R} L_{ird}$ ∀ i ϵ I, ∀ d Є D (10)

$H_i \leq \sum_{d=1}^{7} B_{id} \leq H_i + G_i$ ∀ i ϵ I (11)

$B_{id}=\sum_{t \in T} X_{idt} *8$ ∀ i ϵ I FT, ∀ d ϵ D (12)

$\sum_{t \in T} X_{idt} * 6 \leq B_{id}$ ∀ i ϵ ppt, ∀ d ϵ D (13)

$\sum_{t \in T} X_{idt} * 8 \geq B_{id}$ ∀ i ϵ ppt, ∀ d ϵ D (14)

$\sum_{t \in T} X_{idt} * 8 \geq B_{id}$ ∀ i ϵ Agency, ∀ d ϵ D (15)

$\sum_{r=1}^{t-1} Y_{irs} \leq M*(1-X_{idt})$ ∀iϵI, ∀tЄT, sЄS_d, ∀ d ϵ D (16)

$(r + STs - 1) * Y_{irs} \leq \sum_{t \in T}(t)*X_{idt} + 4*B_{id}$ ∀iϵI, ∀rЄR, ∀ d ϵ D, sЄS_d (17)

$\sum_{s''\neq s'}\sum_{r''=r'}^{r'+ST_{s'}-1} Y_{ir''s''} \leq M*(1-Y_{ir's'})$ ∀ i ϵ I, ∀ r' Є R, ∀ r'' Є R, ∀ s' Є S_d, ∀ s'' Є S_d

(18)

$\sum_{d=g}^{g+6}\sum_{t \in T} X_{idt} \leq 6$ ∀ i ϵ I , gЄ{1 2,…..8} (19)

$\sum_{t \in T} X_{idt} \leq 1$ ∀ i ϵ I, ∀ d ϵ D (20)

$X_{id61}+X_{id65}+X_{i(d+1)1}+X_{i(d+1)29} \leq 1$ ∀ i ϵ I, ∀ d ϵ D-1 (21)

$(r - LB_s) * Y_{irs} \geq 0$ ∀iϵI, ∀sЄS , ∀rЄR (22)

$(UB_s - r) * Y_{irs} \geq 0$ ∀iϵI, ∀sЄS , ∀rЄR (23)

$\sum_{i\in I}\sum_{s\in S}(r * Y_{irs} - k_s) \leq W_s$ ∀rЄR (24)

$\sum_{i\in I}\sum_{s\in S}(k_s - r * Y_{irs})) \leq W_s$ ∀rЄR (25)

Constraint (2) ensures that there should be at least two PSWs available at all times. Constraint (3) ensures at least 1 PSW is assigned to each patient. Constraint (4) ensures the compatibility between the assignment variable, Y_{irs}, and the decision variable capturing visits by each PSW, Z_{ij}. Constraint (5) ensures the available PSWs in each day at each shift who can do service request s should be at least as large as the assigned visit requests to that shift. Constraint (6) ensures that only one PSW is assigned to each service request. If a service request requires two PSWs, it will be split into two separate service requests simultaneously for the same task.

Constraints 7-10 are related to lunch breaks where (7) ensures lunch breaks should not start before the shift, (8) ensures the break should be before the end of the shift, constraint (9) ensures there should be no assignments during the break and constraint (10) ensures that any PSW who works a shift at least 5 hours long should get a break.

Constraints 11-18 relate to availability limitations and the capacity of PSWs. Constraint (11) ensures shift time should be in the defined intervals and constraint (12) requires that full time PSWs get 8-hour shifts. Constraints (13) and (14) ensure PPT and PT PSWs are assigned shifts between 6 to 8 hours in length and constraint (15) ensures Agency PSWs get shifts up to 8 hours per day. Constraints (16) and (17) ensure PSWs are assigned to a service request that is in their shift time and constraint (18) ensures there should be no overlap between two assignments for the same PSW. Constraint (19) prevents PSWs from working more than 6 consecutive days. Constraints (20) and (21) ensure at least 11 hours between two consecutive shifts for each PSW. Constraints (22) and (23) enforce that for each visit request there is a lower and upper bound on when it can be served. Constraints (24) and (25) captures the difference between the preferred and assigned visit times (minimized in the objective function).

5. Proposed Methodology

The HHC scheduling problem is modeled using the above Mixed Integer Linear Program (MILP). MILP provides an efficient method to optimize complex scheduling problems based on defined objectives and constraints over a limited time horizon. Moreover, MILP allows for the inclusion of multiple constraints and objectives, making it adaptable to diverse requirements such as preferences of visit time, patient needs, and shift duration.

Initially, we solved the model using the commercial solver GUROBI and based on a reduced data set. This allowed us to test the model's proper functionality. However, with the full data set we ran into computational intractability forcing us to consider heuristic approaches. Although heuristic approaches do not give an exact optimum solution, they do allow the user to find a practical solution for larger-sized problems. In this thesis, Lagrangian Relaxation and Logic-Based Benders Decomposition were compared in terms of solution time and optimality.

The Lagrangian relaxation technique addresses intractability by taking complex constraints, relaxing them and using penalty terms in the objective to encourage feasibility. By iteratively adjusting the Lagrangian multipliers and solving the relaxed problem, the model gradually improves the solution and reduces constraint violations. This approach makes the problem more manageable and can lead to high-quality solutions for large-scale optimization problems.

The Benders Decomposition approach, on the other hand, decomposes the problem into a master problem and a subproblem, that are solved iteratively. The master problem focuses on the primary decision variables and a simplified version of the constraints, aiming to provide a feasible solution to the reduced problem. The subproblem then checks for specific constraint violations, such as overlaps in assignments in our problem. If violations are found, Benders cuts (constraints) are generated and added to the master problem. This iterative process continues until an optimal solution is found with no constraint violations or until the optimality gap is within a specified tolerance. This decomposition technique effectively handles large-scale optimization problems by breaking them into smaller, more manageable subproblems, leading to efficient and high-quality solutions.

By comparing these two methods, we can determine which approach yields better results in terms of solution quality and computational efficiency for this specific optimization problem.

5.1 Basic Concepts of Lagrangian Relaxation

Optimization Problem Formulation

In a typical optimization problem, the aim is to minimize or maximize an objective function subject to a set of constraints. The problem can be formulated as follows:

Minimize f(x)

subject to:

$g_i(x) \leq 0 \quad i=1,\ldots,m$

$h_j(x) = 0 \quad j=1,\ldots,p$

Here, f(x) is the objective function, $g_i(x)$ represents inequality constraints, and $h_j(x)$ represents equality constraints.

Lagrangian Function

The Lagrangian function is constructed by incorporating (a subset of) the constraints into the objective function using Lagrangian multipliers (λ_i for inequality constraints):

L(x, λ)=f(x)+$\sum_i \lambda_i * g_i(x)$

subject to:

h_j(x)=0 j=1,…,p

The Lagrangian multipliers (λ_i) act as penalty coefficients that quantify the extent to which relaxed constraints are violated.

Lagrangian Dual Problem

The Lagrangian dual problem involves two main steps. First, the original optimization problem is transformed by incorporating the constraints into the objective function using Lagrangian multipliers, creating the Lagrangian function. The objective is then to minimize this Lagrangian function. Once this minimization is achieved, the next step is to maximize the resulting function with respect to the Lagrangian multipliers. This maximization process seeks the best possible lower bound for the original problem, offering insights into the problem's structure and potential optimality. This dual problem provides a lower bound for the original optimization problem (primal problem), and the difference between the primal and dual solutions, known as the duality gap, indicates the quality of the approximation.

Advantages and Challenges

Lagrangian relaxation simplifies complex problems by relaxing constraints, making the problem more tractable. The technique allows for iterative improvements of the solution quality, leading to near-optimal solutions. Lagrangian relaxation can be applied to various types of optimization problems across different fields.

However, the presence of a duality gap indicates that the relaxed problem may not provide an exact solution to the original problem. Efforts are needed to minimize this gap. Ensuring convergence to an optimal solution requires careful tuning of parameters such as the step size and initial values of the Lagrangian multipliers. For large-scale problems, the iterative process can be computationally intensive, necessitating efficient algorithms and heuristics.

Iterative Solution Process

The iterative process used to solve the relaxed problem involves the following steps:

- Set initial values for the Lagrangian multipliers (λ_i).
- Solve the relaxed problem for the current values of λ_i to obtain the decision variables (x).
- Adjust the Lagrangian multipliers based on the constraint violations. This can be done using subgradient methods, where the multipliers are updated as follows:

  $$\lambda_i^{(k+1)}=\lambda_i^{(k)}+\alpha^{(k)}g_i(x^{(k)})$$

  Here, $\alpha^{(k)}$ is the step size at iteration k, and $x^{(k)}$ represents the decision variables at iteration k.

- Repeat the optimization and update steps until convergence criteria are met, such as a small duality gap or a maximum number of iterations.

Augmented Lagrangian Relaxation (ALR)

Augmented Lagrangian Relaxation (ALR) enhances Lagrangian Relaxation by introducing an additional positive penalty term on the constraint violations. While the standard Lagrangian already penalizes constraint violations through linear terms, the positive penalty in ALR helps to better enforce feasibility by discouraging large violations more strongly. This added term also improves the convergence properties of the optimization algorithm, preventing oscillations and leading to more stable and efficient solutions.

5.2 ALR Applied to PSW Scheduling

In our setting, we relax the following constraint:

$\sum_{s" \neq s'} \sum_{r"=r}^{r'+ST_{s'}-1} Y_{ir"s"} \leq M * (1 - Y_{ir's'})$ ∀ i ϵ I, ∀ r' Є R, ∀ r" Є R, ∀ s' ЄS_d, ∀ s" Є S_d

"There should be no overlap between two assignments for each PSW"

This constraint was chosen because its impact on solution time was observed by incrementally adding each constraint to the model and solving it.

We then implement the following instance of the algorithm outlined above.

1. Initialization of Lagrangian Multipliers:

   The Lagrangian multipliers (λ) are uniformly initialized across all combinations of PSWs (i), days (d), times (r), and service requests (s) because the constraints are expected to have similar impact on the objective function. These multipliers are set to an initial value similar to other objective function weights to facilitate the exploration of the solution space.

   $\lambda_{idr's'}$= λ ∀ i ∈ I, ∀ d ∈ D, ∀ r' ∈ R, ∀ s' ∈ S_d

2. Subgradient Optimization:

   $\lambda_{idr's}^{(k)} = \lambda_{idr's'}^{(k-1)} + \boldsymbol{\alpha}*$ (violation)

- Initial step size (α): Determines the initial size of the step taken in each iteration to adjust the Lagrangian multipliers. A fixed initial step size is chosen to quickly steer the optimization process toward a feasible solution. This approach enables faster identification of regions in the solution space where constraints are less violated.
- Iterations (k): Set the upper limit on the number of iterations for the optimization process.
- Violation threshold: Establish a threshold for acceptable constraint violations.

3. Identifying Potential Violations:

   For each combination of PSW (i), day (d), time (r), and service request (s), the model calculates potential violations of the relaxed constraints. This involves computing the left-hand side (LHS) and right-hand side (RHS) of the relaxed constraints.

   - LHS Calculation:

     $$\mathrm{LHS}=\sum_{s''\neq s} \sum_{r''=r'}^{r'+S_{s'}-1} Y_{ir''s''}$$

   - RHS Calculation:

     $$\mathrm{RHS}=M*(1-Y_{ir's'})$$

   - Violation Expression:

     $$\text{violation}=\text{LHS}-\text{RHS}$$

   - Non-Negative Violation:

     A non-negative variable $violation_{idr's}$ is introduced to ensure that the violation is non-negative:

     $$violation_{idr's} \geq \text{violation}$$

- Storing Violation Parameters:

  To manage and store the identified violations, the parameters are appended to a list named violation_params. Specifically, for each combination of i, d, r'and s', the tuple (i, d, r, s', $\lambda_{idr's}$, $violation_{idr's'}$)) is stored in violation_params.

The objective function is defined as the sum of the original objective components and the penalties for constraint violations.

Objective Function:

$$\text{Min } \left(w_1\left(\frac{\sum_{s\in S} W_s}{\mathrm{W_{max}}}\right) + w_2\left(\frac{\sum_{i\in (FT,PPT,PT)}\sum_{d\in D} C_i * B_{id} + \sum_{i\in (Agency)}\sum_{d\in D} C_i * B_{id}}{B_{max}}\right) + w_3\left(\frac{\sum_{i\in I}\sum_{j\in J} Z_{ij}}{Z_{max}}\right) + \sum_{i\in I}\sum_{d\in D}\sum_{r'\in R}\sum_{s'\in \mathrm{SD}} \lambda_{idr's'} * violation_{idr's'}\right)$$

5.2.1 Using LR Results for Warm Start

By applying Lagrangian Relaxation, we obtain a set of Lagrangian multipliers and an approximate solution that minimizes constraint violations but does not necessarily eliminate them.

The solution obtained from LR, though approximate, provides a feasible starting point for the main optimization problem. This solution is closer to the optimal solution compared to an arbitrary or naive initial solution. Using the LR solution as a warm start significantly reduces the solution time of the main problem.

By integrating the LR results as a warm start, we leverage the approximate solutions to accelerate the overall optimization process, aiming to find an improved solution within an acceptable time frame. This approach helps ensure that hard constraints are effectively met while enhancing computational efficiency, even if the solution is not guaranteed to be strictly optimal.

5.3 Basic Concepts of Logic-Based Bender's Decomposition

Benders Decomposition is a powerful algorithm used to solve large-scale optimization problems. Logic-based Bender's decomposition (LBBD) extends traditional Benders decomposition, making it applicable to a broader range of combinatorial optimization problems. This method is especially useful in scheduling as it allows the integration of mixed-integer programming (MIP) and constraint programming (CP), leveraging the strengths of both methods. While it shares similarities with methods like Column Generation and Cutting Plane techniques, it offers distinct advantages. Column Generation dynamically generates variables, making it effective for linear and large-scale problems but less suitable for complex, non-linear, or integer constraints. Cutting Plane methods refine the LP relaxation by adding cuts, tightening the feasible region but potentially slowing convergence in complex cases. In contrast, Logic-Based Benders Decomposition typically requires fewer iterations by leveraging the logical structure of the problem, generating cuts that directly influence the master problem's feasible region, leading to faster convergence. [28]

The technique divides the problem into two interrelated problems: the **master problem** and the **subproblem**.

Master Problem:

- Includes a subset of the original problem's variables and constraints, typically the integer variables.
- Solving the master problem provides a (possibly infeasible) solution to the original problem.

Subproblem:

- Includes the remaining variables and constraints that were not included in the master problem.
- The subproblem checks the feasibility of the master problem's solution and provides feedback (Benders cuts) to improve the master problem's solution.

Advantages and Challenges

Benders Decomposition breaks down a large, complex problem into smaller, more manageable subproblems. This decomposition facilitates solving problems that are otherwise intractable. By focusing on the master problem and iteratively refining it with cuts from the subproblem, Benders Decomposition can lead to significant computational savings, especially for large-scale optimization problems. It is thus scalable for large-scale linear and mixed-integer programming problems.

There are, however, some challenges. For instance, generating effective Benders cuts can be complex and requires careful formulation to ensure convergence and efficiency. Moreover, the convergence to the optimal solution can sometimes be slow, depending on the nature of the problem and the effectiveness of the cuts and finding a good initial feasible solution for the master problem can be challenging and critical for the efficiency of the algorithm. Thus, implementing Benders Decomposition requires a deep understanding of the problem structure and the decomposition technique, making it more challenging to implement compared to other methods.

5.4 LBBD Applied to PSW Scheduling

Logic-Based Bender's decomposition is applied in this problem to effectively manage the complexity of the overlap violation constraints. Since these constraints add significant computational intensity, they are separated out into a subproblem. This allows the master problem to focus on the simpler aspects of the optimization, while the subproblem deals with the more intricate feasibility checks related to overlaps.

Master Problem

The master problem involves all the variables and constraints except the overlap violation:

The objective function, as explained in Section 4.5, is to minimize

$$\text{Min } \left(w_1\left(\frac{\sum_{s\in S} W_s}{\mathrm{W_{max}}}\right) + w_2\left(\frac{\sum_{i\in (FT,PPT,PT)} \sum_{d\in D} C_i * B_{id} + \sum_{i\in (Agency)} \sum_{d\in D} C_i * B_{id}}{B_{max}}\right) + w_3\left(\frac{\sum_{i\in I} \sum_{j\in J} Z_{ij}}{Z_{max}}\right)\right.$$

Subproblem

The subproblem checks for overlap violations. Given a solution $\mathrm{Y_{irs}}$ from the master problem, the subproblem is formulated as:

Minimize θ

Subject to:

$$\sum_{\mathrm{s"\neq s}} \sum_{\mathrm{r"=r}}^{\mathrm{r'+ST}_{s'}-1} \mathrm{Y_{ir"s"}} - \mathrm{M} * (1 - \mathrm{Y}_{\mathrm{ir's'}}) \leq \theta \quad \forall \text{ i} \in \text{I}, \forall \text{ r'} \in \text{R}, \forall \text{ r"} \in \text{R}, \forall \text{ s'} \in S_d, \forall \text{ s"} \in S_d$$

Where θ is a continuous variable representing the amount of overlap violation.

5.4.1 Benders Decomposition Process

Step 1: Solve the Master Problem

Solve the master problem to get a candidate solution Y and objective value. This step provides an initial solution that is likely infeasible for the full problem due to overlaps.

Step 2: Solve the Subproblem

For the given Y, solve the subproblem to check for overlap violations. The objective of the subproblem represents the extent of any overlaps.

Step 3: Generate Benders Cuts

If the subproblem finds overlap violations (i.e., θ >0), generate a Benders cut to eliminate this violation in future iterations. The cut is of the form:

$$\sum_{s"\neq s'} \sum_{r"=r}^{r'+ST_{s'}-1} Y_{ir"s"} \leq M * (1 - Y_{ir's'}) \qquad \forall\ i \in I,\ \forall\ r' \in R,\ \forall\ r" \in R,\ \forall\ s' \in S_d,\ \forall\ s" \in S_d$$

These cuts are added to the master problem as new constraints.

Step 4: Iterate

Iterate between solving the master problem and the subproblem until:

- No more overlap violations are found.
- And the optimality gap (difference between the upper and lower bounds) is within a specified tolerance.

6. Input Data

NIL provided the following data on the PSWs' union rules and client tasks.

6.1 PSW Information:

- 17 PSWs: 8 full-time (FT), 1 permanent part-time (PPT), and 8 part-time (PT)
- Compatibility of each PSW with service requests
- FT PSWs: 80-88 hours every two weeks, with 8-hour shifts
- PPT and PT PSWs: 60-74 hours every two weeks, with a minimum 5-hour shift

- 11-hour break between shifts and a maximum of 6 consecutive working days
- Typical shifts: 12am-8am, 7am-1pm, 7am-3pm, 3pm-11pm, 4pm-12am
- Two PSWs must be on duty at all times

6.2 Client Information:

- 16 clients currently require between 1 to 17 services (clients who need 2 PSWs, considered as two different requests at the same time) per day.
- Tasks for each client include the day, time (±15 minutes adjustment allowed), duration, and the number of PSWs required

These parameters are included in the mathematical model and the following preprocessing adjustments were made:

- The original data listed all client (patients) service requests in tables by day. This data was consolidated into one table to create a comprehensive list of all services required over a two-week period.
- The first column in Table 3 represents the service request's number, which are 1-1385.
- Column 2 in Table 3 provides the patient number associated with the service request.
- Column 3 provides the day of service (the days were adjusted to a 1-14 format).
- The preferred visit time as well as columns for upper and lower bounds of service start times are found in columns 4-6 respectively. We consider a maximum adjustment of ±15 minutes. This allows the model to utilize a range of available times to start each task.

- Deterministic service durations are provided in column (7) with numbers ranging from 1 to 4. Each number represents a multiple of 15-minute increments. For instance, a service duration of 2 means a 30-minute service request.

- The compatibility of each PSW with each service request was converted to values of 0 or 1 (Table 4), enabling Python to import this data as integers instead of strings.

- The allowed working hours per two-week period for each type of PSW (full-time, permanent part-time, part-time, and agency) were added to a table defining upper and lower bounds.

These adjustments were made to facilitate communication between Excel and Python and to organize the data to fit the mathematical model. For example, the first row of Table 3 shows that service request 1 for patient 1 on day 1 is 15 minutes long and should preferably be provided at time 3 (00:45 am) but can be started anywhere from 00:30 am and 1:00 am.

Table 3: Input data Sample

Service request(S)	Patient(J)	Day(D)	Preferred visit time (k)	Lower bound for visit time (LB)	Upper bound for visit time (UB)	Service type (ST)
1	1	1	3	2	4	1
206	3	2	71	70	72	2
318	4	2	9	8	10	1

Table 4 indicates the compatibility between PSWs and patients. For instance, PSW 4 cannot visit patient 5.

Table 4: PSW compatibility sample

i \ j	1	2	3	4	5
1	1	1	1	0	1
2	1	0	1	1	1
3	1	1	1	1	1
4	1	1	1	1	0

6.3 Assumptions

There are a number of assumptions upon which our model relies. First, the start time of the shifts for PSWs are restricted to a subset of all possible start times but the end varies depending on FT, PPT, PT or agency. Secondly, there is no priority to assign FT, PPT and PT PSWs at each shift other than meeting the required hours for each type. Thirdly, patients have been informed they will be visited at the scheduled time and there is no cancellation and rescheduling in the planning period. Fourthly, the traveling time between clients is considered negligible since all of them are in the same complex. Finally, service times are assumed to be discrete and deterministic for easier calculation.

7. Results of LBBD and ALR

In this section, we present a comparison of the results obtained using Logic-Based Benders Decomposition (LBBD) and Augmented Lagrangian Relaxation (ALR) when applied to the MIP model. We present a small-scale case that can be solved to optimality and then a more realistically-scaled case populated by the data from NIL.

7.1 Solution Quality

In the small-scale case, there are 9 patients, 11 PSWs and 88 service requests over a 2-week period and the other parameters are defined in the mathematical model and input data sections and are the same as for the large scale case, which are mentioned in sections 4.1, 4.2, 6.1 and 6.2.

Table 5:Small-scale result comparation

Method	**Best objective**	**Best bound**	**Gap (%)**	**Time (s)**
Lagrangian Relaxation	1929.38	1929.30332	0.0037%	10
Benders Decomposition	1929.42	1929.30332	0.0062%	13

The **gap (%)** represents the optimality gap, which measures the difference between the best objective value found by the proposed method and the best possible bound on the objective value across all unexplored nodes in the branch-and-bound tree.

$$\text{Gap (\%)}=\left(\frac{\textbf{Best objective}-\textbf{Best bound}}{\textbf{Best bound}}\right)*100$$

The results provided in Table 5 indicate that both methods can find near-optimal solutions efficiently. However, LR marginally outperforms BD in terms of both the solution quality and computational time for the small-scale problem, providing confidence in its effectiveness when applying these methods to larger, real-sized datasets.

While both methods effectively addressed the small-scale case, the results for the larger real-sized case favor the LBBD approach over LR, as shown in Table 6. Moreover, as shown in Table 6, the best bounds obtained from these heuristic methods—2757.4 for ALR and 2757.5 for LBBD—produced a 1.67% gap compared to the original problem's best bound of 2711.4.

Table 6:Result comparison

Method	**Best objective**	**Best bound**	**Gap (%)**	**Time (s)**
Lagrangian Relaxation	3026.9	2757.4	8.906%	63600
ALR as warm start	2771.2	2711.4	2.158%	12000
Benders Decomposition	2762.9	2757.5	0.195%	75600

This relatively small gap indicates that the heuristics are performing well, particularly given the complexity of the problem and the significant reduction in computational time—to less than a day—while still maintaining strong performance.

7.2. Convergence Rate

Convergence rate is crucial for the efficiency of an algorithm with faster convergence implying a quicker attainment of satisfactory solutions.

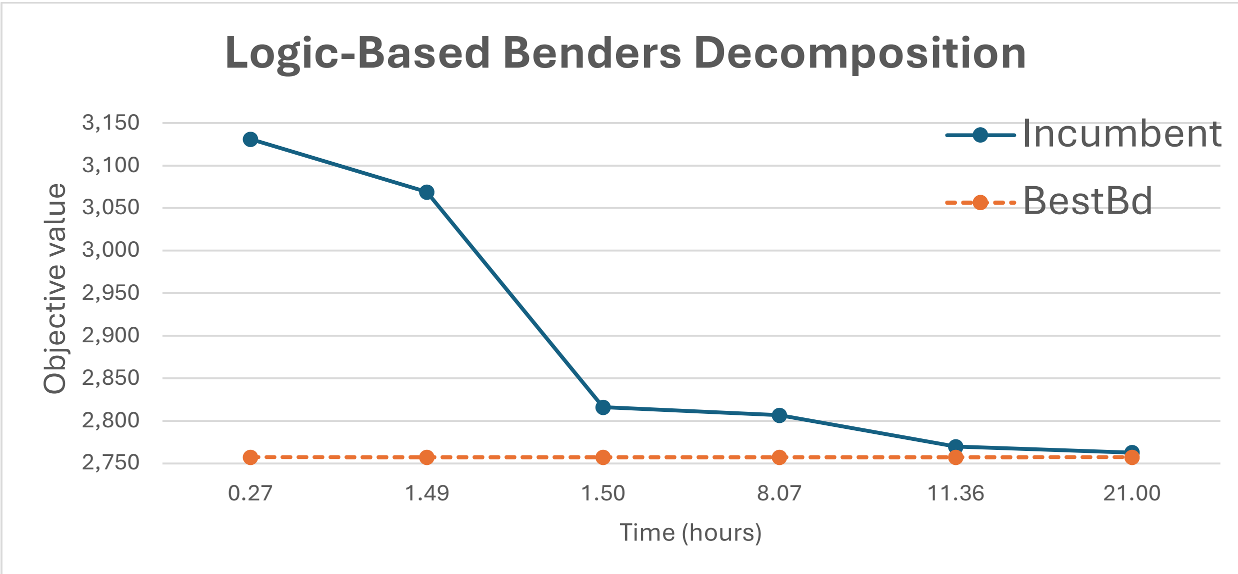

Figure 1 :Benders Decomposition convergence

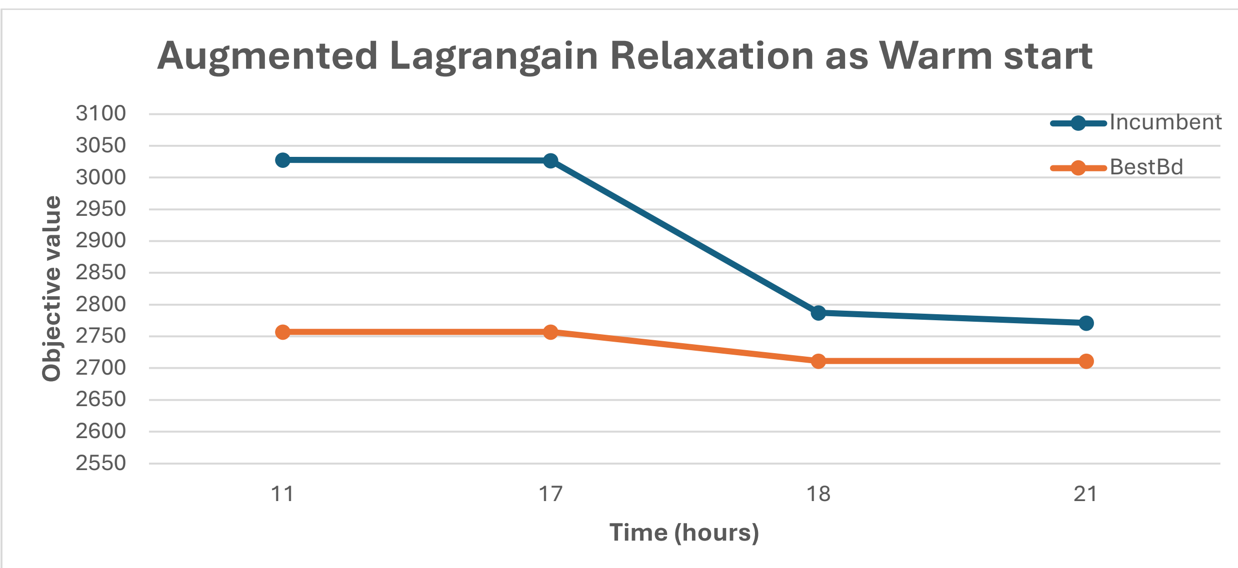

Figure 2:Augmented Lagrangian relaxation convergence

Figures 1 and 2 illustrate that LBBD converges to a high-quality solution much faster than ALR, despite both methods having the same total runtime of 21 hours. A portion of ALR's time (17 to 21 hours) is related to using the warm start in the original problem. While the warm start helps ALR reduce the gap early on, it could not reach a solution as close to optimal as LBBD within the same time frame. This accelerated convergence makes LBBD a more efficient choice to solve large-scale scheduling problems. LBBD also significantly cuts down the memory required to solve the problem, staying within reasonable limits even for larger problem sizes.

7.3 Feasibility Handling

Feasibility refers to the ability of an algorithm to find solutions that satisfy all problem constraints. LBBD enforces feasibility through its decomposition approach, ensuring that all constraints are satisfied in each subproblem iteration. In contrast, ALR relies on penalty parameters to manage infeasibility, which leads to solutions that temporarily violate constraints until the penalty terms adequately penalize such violations. As a result, LBBD provides more reliable feasibility handling without the need for extensive parameter tuning.

By examining LBBD and LR across solution quality, convergence rate, feasibility handling and scalability, we gain an understanding of their respective strengths and limitations in addressing the MILP scheduling problem. Since the Benders Decomposition method provides better outcomes in equal time, we provide a deeper dive into the results from running it below.

In the objective function, LBBD was able to schedule PSWs to meet patient service requests at their preferred time 25% of the time over a two-week period, with the remaining 75% scheduled within a 15-minute difference from the preferred time. Moreover, the LBBD approach yielded a solution where only one agency PSW was used over two weeks, marking a significant improvement compared to NIL's current schedule.

8. Sensitivity Analysis of Weight Parameters

In optimization, sensitivity analysis is a crucial step to understand how changes in model parameters affect the outcome. By varying key parameters, we can assess the robustness of the solution and identify which parameters have the most significant impact on the objective function and decision variables. In this study, we performed a sensitivity analysis on the weight parameters W_1, W_2, W_3 in the objective function that determine the relative importance of visiting at the preferred time, reducing the number of agency PSWs and providing continuity of care within the scheduling model respectively.

To conduct the sensitivity analysis, we varied each weight parameter while keeping the other parameters constant (Baseline). The values of Weights were tested at four levels: the baseline, 50% increase, 100% increase, and 25% decrease. For each value of weights, the optimization model was rerun, and the results collected, focusing on the following key metrics:

- The objective function value.
- Visiting at the preferred time
- The number of working hours of agency PSWs.
- The average number of PSWs allocated to a patient.

8.1 W_1 :Visiting at the preferred time

The results of the sensitivity analysis are summarized in the following tables, which show how varying the weight parameter W_1 affects different aspects of the scheduling model.

Table 7:the objective function values for each tested value of w_1

Setting	*best objective*	*Best bound*	*gap*	*Time (S)*
25% Decrease	2751.7	2746.3	0.20%	75600
Baseline	2762.9	2757.5	0.20%	75600
50% Increase	2785.2	2779.8	0.20%	75600
Doubled W1	2809.8	2802.1	0.28%	75600

Increasing the value of W_1 had a small impact on the objective function value. When W_1 was increased by 50%, the best objective value increased slightly from 2762.9 to 2785.2, with the gap between the best objective and best bound remaining constant at 0.20%. When W_1 was doubled, the best objective value increased further to 2809.8, and the gap increased slightly to 0.28%. Conversely, reducing W_1 by 25% resulted in a minor decrease in the best objective value to 2751.7, with no change in the gap. This indicates that while increasing or decreasing W_1 slightly affects the problem's difficulty, the model remains robust to these changes.

Table 8:The gap between preferred and real visit time

Setting	Total service requests in two weeks	%	Assignments on exact preferred time	%	Assignments with 15 min tolerance	%
25% Decrease	1385	100.00%	345	24.91%	1040	75.09%
Baseline	1385	100.00%	345	24.91%	1040	75.09%
50% increase	1385	100.00%	346	24.98%	1039	75.02%
Doubled W1	1385	100.00%	354	25.56%	1031	74.44%

As W_1 increased, the percentage of assignments made at the exact preferred time improved slightly. With a 50% increase in W_1 , the percentage of assignments at the preferred time rose from

24.91% to 24.98%. When W_1 was doubled, this percentage increased further to 25.56%. Conversely, reducing W_1 by 25% resulted in no change in the percentage of assignments at the preferred time, remaining at 24.91%.

The total working hours assigned to agency PSWs remained constant at 1 hour over the two-week period, regardless of changes in W_1 .

Table 9:the allocation of PSWs to patients

Patient ID	Number of service request per patient	Number of different PSW Visited in two weeks -25% Decrease	Number of different PSW Visited in two weeks-Baseline	Number of different PSW Visited in two weeks-50% increase W1	Number of different PSW Visited in two weeks-Doubled W1
1	185	17	17	17	17
2	9	3	3	3	3
3	112	13	13	13	14
4	148	15	15	15	14
5	70	12	12	12	12
6	15	5	5	5	5
7	127	16	16	16	16
8	232	16	16	16	17
9	52	11	11	11	11
10	14	5	5	5	6
11	164	14	14	14	14
12	57	9	9	9	9
13	64	13	13	13	13
14	25	8	8	8	8
15	20	7	7	7	7
16	91	14	14	14	14
Average	86.57	11.13	11.13	11.13	11.25

The allocation of PSWs to patients showed minor variations across different settings. With a 50% increase in W_1 , the average number of different PSWs visiting each patient over two weeks remained unchanged at 11.13. When W_1 was doubled, the average increased slightly to 11.25. Reducing W_1 by 25% resulted in no change to the average number.

The sensitivity analysis demonstrates that both increasing and decreasing W_1 had a limited impact on the model's key metrics. These findings suggest that the model is relatively robust to variations in W_1 .

8.2 W_2 : Reduction in the number of Agency PSWs employed

The results of the sensitivity analysis are summarized in the following tables and figures, which show how varying the weight parameter W_2 affects different aspects of the nurse scheduling model.

Table 10:the objective function values for each tested value of W_2

SETTING	BEST OBJECTIVE	BEST BOUND	GAP	TIME (S)
25% DECREASE	2083.7	2079.3	0.21%	75600
BASELINE	2762.9	2757.5	0.20%	75600
50% INCREASE	4125.0	4113.8	0.27%	75600
DOUBLED W_2	5507.7	5470.1	0.68%	75600

Increasing W_2 resulted in an increase in the objective function value. With a 50% increase in W_2, the best objective value increased from 2762.9 to 4125.0, and the gap between the best objective and best bound increased from 0.20% to 0.27%. When W_2 was doubled, the best objective value rose further to 5507.7, and the gap widened to 0.68%. Conversely, a 25% decrease in W_2 reduced the best objective value to 2083.7, with a gap of 0.21%. These results suggest that increasing W_2 makes the problem more challenging to solve optimally, reflecting difficulty in further reducing the agency usage. Decreasing W_2 reduces the focus on agency usage, making it relatively easier to solve.

Table 11:The gap between preferred and real visit time

SETTING	TOTAL SERVICE REQUESTS IN TWO WEEKS	%	ASSIGNMENTS ON EXACT PREFERRED TIME	%	ASSIGNMENTS WITH 15 MIN TOLERANCE	%
25% DECREASE	1385	100.00%	375	27.08%	1010	72.92%
BASELINE	1385	100.00%	345	24.91%	1040	75.09%
50% INCREASE	1385	100.00%	344	24.84%	1041	75.16%
DOUBLED W2	1385	100.00%	344	24.84%	1041	75.16%

As W_2 increased, the percentage of assignments made at the exact preferred time remained relatively stable, decreasing slightly from 24.91% at the baseline to 24.84% for both the 50% increase and doubled W_2 settings. On the other hand, a 25% decrease in W_2 increased the exact preferred time assignments to 27.08%. This suggests that decreasing W_2 improves the ability to meet patient preferences for visit times while increasing it does not significantly impact this metric. The total working hours assigned to agency PSWs remained constant at 1 hour over the two-week period when increasing the weight. However, the working hours increased to 2 hours when the weight is decreased. This finding suggests that completely removing agency PSW usage is impossible within the constraints of the model.

Table 12:the allocation of PSWs to patients' changes

Patient ID	Number of service request per Patient	Number of different PSW Visited in two weeks-25% Decrease	Number of different PSW Visited in two weeks-Baseline	Number of different PSW Visited in two weeks-50% increase W2	Number of different PSW Visited in two weeks-Doubled W2
1	185	16	17	18	18
2	9	5	3	6	3
3	112	15	13	14	13
4	148	17	15	14	16
5	70	11	12	11	10

6	15	8	5	4	4
7	127	15	16	16	15
8	232	17	16	16	17
9	52	13	11	12	11
10	14	6	5	5	5
11	164	14	14	14	15
12	57	11	9	7	10
13	64	15	13	12	14
14	25	8	8	6	6
15	20	6	7	7	6
16	91	16	14	16	16
Average	86.56	12.06	11.13	11.13	11.19

The allocation of PSWs to patients showed minor changes with variations in W_2 . With a 50% increase in W_2 , the average number of different PSWs visiting each patient over two weeks remained unchanged at 11.13. When W_2 was doubled, the average increased slightly to 11.19. A 25% decrease in W_2 increased the average number of different PSWs to 12.06. This suggests that decreasing W_2 slightly improves the continuity of care by assigning fewer PSWs to patients, whereas increasing W_2 reflects a trade-off between reducing agency use and maintaining consistent patient assignments.

Overall, the model shows relative robustness to changes in W_2. A higher W_2 increases the challenge in reaching an optimal solution while balancing the need for reduced agency usage against other operational and patient-centered metrics. However, decreasing W_2 improves continuity of care and patient preference satisfaction at the expense of higher agency usage.

8.3 W_3 : Continuity of care

The results of the sensitivity analysis are summarized in the following tables, which demonstrate how varying the weight parameter W_3 affects different aspects of the scheduling model.

Table 13: the objective function values for each tested value of W_3

Setting	*best objective*	*Best bound*	*gap*	*Time (S)*
25% Decrease	2771.8	2757.4	0.52%	75600
Baseline	2762.9	2757.5	0.20%	75600
50% increase	2768.9	2757.6	0.41%	75600
Doubled W3	2773.1	2757.7	0.55%	75600

As W_3 increased, the best objective value also increased, indicating a shift in the model's priorities towards continuity of care. The gap between the best objective and the best bound grew from 0.20% at baseline to 0.41% with a 50% increase in W_3, and to 0.55% when W_3 was doubled. Conversely, a 25% decrease in W_3 yielded a smaller gap of 0.52%, suggesting a marginal reduction in difficulty. Runtime remained constant across all settings.

Table 14: The gap between preferred and real visit time

Setting	Total service requests in two weeks	%	Assignments on exact preferred time	%	Assignments with 15 min tolerance	%
25% Decrease	1385	100.00%	345	24.91%	1040	75.09%
Baseline	1385	100.00%	345	24.91%	1040	75.09%
50% increase	1385	100.00%	390	28.16%	995	71.84%
Doubled W2	1385	100.00%	390	28.16%	995	71.84%

With a 50% increase in W_3, the percentage of assignments at the exact preferred time increased from 24.91% to 28.16%. When W_3 was doubled, the percentage remained at 28.16%. However, reducing W_3 by 25% maintained the baseline value of 24.91%. These results indicate that prioritizing continuity of care with higher W_3 values does not conflict with meeting patients' preferred times but does not improve this alignment when W_3 decreases.

Agency PSW working hours increased from 1 hour to 6 hours over the two-week period when W_3 was increased by 50% and doubled. A 25% decrease in W_3 retained the baseline usage, suggesting that higher W_3 values demand more flexibility from agency PSWs.

Table 15:the allocation of PSWs to patients changes as w3 modified

Patient ID	Number of service request per Patient	Number of different PSW Visited in two weeks-25% Decrease W3	Number of PSWs per client over two weeks-Original	Number of different PSW Visited in two weeks-50% increase W3	Number of different PSW Visited in two weeks-Doubled W3
1	185	16	17	17	17
2	9	4	3	3	2
3	112	13	13	13	12
4	148	13	15	14	15
5	70	13	12	12	10
6	15	6	5	5	5
7	127	15	16	15	16
8	232	17	16	16	17
9	52	11	11	11	9
10	14	4	5	5	4
11	164	15	14	15	15
12	57	11	9	11	11
13	64	13	13	11	11
14	25	7	8	7	7
15	20	7	7	7	6
16	91	15	14	15	15
Average	86.56	11.25	11.13	11.06	10.75

As W_3 increased, the average number of PSWs visiting each patient over the two weeks decreased from 11.13 to 11.06 with a 50% increase in W_3, and further to 10.75 when W_3 was doubled. A 25% decrease, however, slightly increased this figure to 11.25.

Overall, the model demonstrates relative robustness to changes in W_3, with moderate impacts on the objective function value and PSW allocation. Higher W_3 values successfully promote continuity of care but introduce trade-offs, such as increased agency PSW usage and a slight increase in scheduling complexity, as reflected in the widening optimality gap. Reducing W_3, on the other hand, modestly impacts both continuity and scheduling efficiency.

The results of the sensitivity analysis indicate that the model is relatively robust to changes in the weight parameters W_1, W_2, and W_3, showing only slight variations in key metrics despite significant adjustments to the weights. This robustness can be attributed to several factors. Firstly, the objective functions may be highly correlated (particularly "meeting at preferred time" and "continuity of care"), meaning that improving one objective also tends to improve the others, thereby dampening the impact of changing individual weights. Secondly, the constraints within the MILP model may tightly limit the feasible region, restricting the degree to which the solution can change even when weights are adjusted. These factors together suggest that the model's structure inherently balances multiple objectives in a way that is not easily disrupted by significant changes in weight parameters, leading to a consistent performance across different scenarios.

9. Conclusion

The importance and relevance of the research problem lie in its potential to significantly contribute to the advancement of knowledge in the field of scheduling, particularly within the context of home healthcare settings like Nucleus Independent Living.

Efficient nurse scheduling is critical to ensure that patient care needs are met effectively despite limited resources. By developing a robust scheduling methodology using a Mixed-Integer Linear Programming (MILP) model, this research optimized resource allocation, leading to improved patient outcomes and enhanced operational efficiency. The MILP framework allowed for the integration of complex constraints and objectives, making it well-suited for the nurse scheduling problem.

The proposed solution methods included Logic-Based Benders Decomposition (LBBD) and Augmented Lagrangian Relaxation (ALR), both of which were applied to solve the MILP model efficiently. These methodologies addressed computational challenges associated with large-scale problems while ensuring high-quality solutions. Effective scheduling plays a pivotal role in ensuring timely and appropriate delivery of care to patients. By incorporating patient visit requirements, acuity levels, and preferences into the scheduling process, this research can facilitate better coordination of care, leading to improved patient satisfaction, health outcomes, and overall quality of care.

Nurse or PSW scheduling must comply with various regulatory standards and guidelines, including staffing ratios, labor laws, and healthcare regulations. By using MILP and employing LBBD and ALR as solution methods, this research helps healthcare organizations like Nucleus Independent Living maintain compliance while optimizing key performance metrics. The study highlighted the strengths and limitations of these methods, demonstrating that LBBD, in particular, delivered superior results for real-world scenarios.

By applying these advanced optimization techniques to address the complexities of nurse scheduling in home healthcare settings, our research showcased the practical application of mathematical modeling and innovative solution strategies. This work advances the methodological toolkit available to healthcare practitioners and researchers alike.

Last but not least, the findings of the research may have broader implications beyond Nucleus Independent Living, with potential applicability to other home healthcare agencies with similar care settings. By showcasing how well the suggested scheduling approach performs in various situations, the study provides a foundation for making scheduling best practices more widely applicable and adaptable across the healthcare sector.

10. References

[1] Scheffler RM, Arnold DR, "Projecting shortages and surpluses of doctors and nurses in the OECD: what looms ahead," *Health Econ Policy Law,* pp. 274-290, 2019.

[2] Christian Fikar, Patrick Hirsch, "Home health care routing and scheduling: A review," *Computers & Operations Research,* pp. 86-95, 2017.

[3] A.T. Ernst, H. Jiang, M. Krishnamoorthy , D. Sier, "Staff scheduling and rostering: A review of applications, methods and models," *European Journal of Operational Research,* p. 3–27, 2004.

[4] Edmund K. Burke, Patrick de Causmaecker, Greet Vanden Berghe, Hendrik Van Landeghem, "The state of theart of nurse rostering," *Journal of Scheduling,* p. 441–499, 2004.

[5] P. D. C. Stefaan Haspeslagh, "The first international nurse rostering competition 2010," in *The first international nurse rostering competition*, Belfast, 2012.

[6] Ceschia, S., Dang, N., "The Second International Nurse Rostering Competition," *Ann Oper Res 274,* p. 171–186, 2019.

[7] Mischek, F., Musliu, N, " Integer programming model extensions for a multi-stage nurse rostering problem," *Ann Oper Res 275,* p. 123–143, 2019.

[8] Elín Björk Böðvarsdóttir, Niels-Christian Fink Bagger, "A flexiblemixed integer programming-based system for real-world nurse rostering," *Journal of Scheduling,* p. 59–88, 2022.

[9] Jia Guo, Jonathan F. Bard, "A column generation-based algorithm for midterm nurse scheduling with specialized constraints, preference considerations, and overtime," *Computers and Operations Research,* vol. 138, p. 105597, 2022.

[10] Celia A. Glass , Roger A. Knight, "The nurse rostering problem: A critical appraisal of the problem structure," *European Journal of Operational Research,* p. 379–389, 2010.

[11] StevenJ.M.denHartog, HanHoogeveena, "On the complexity of Nurse Rostering problems," *Operations Research Letters,* p. 483–487, 2023.

[12] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society,* vol. 64, no. 5, pp. 275-278, 1958.

[13] S. &. V. L. Boyd, Convex Optimization, Cambridge: Cambridge University Press, 2004.

[14] Edmund K. Burke1, Tim Curtois2, "New approaches to nurse rostering benchmark instances," *European Journal of Operational Research,* p. 71–81, 2014.

[15] Santos, H.G., Toffolo, T.A.M., Gomes, R.A.M, "nteger programming techniques for the nurse rostering problem," *Ann Oper Res 239,* p. 225–251, 2016.

[16] F. Guessoum, S. Haddadi & E. Gattal, "Simple, Yet Fast and Effective Two-Phase Method for Nurse Rostering," *American Journal of Mathematical and Management Sciences,* pp. 1-19, 2019.

[17] Christos Valouxis, Christos Gogos, George Goulas, "A systematic two phase approach for the nurse rostering problem," *European Journal of Operational Research,* p. 425–433, 2012.

[18] D. e. al., Column Generation, Springer Science & Business Media., 2005.

[19] L. a. Desrosiers, "Selected topics in column generation," *Operations Research,* pp. 1007-1023, 2005.

[20] M. L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science-INFORMS,* vol. 50, pp. 1861-1871, 2004.

[21] A. M. F.-F. e. al., "A Lagrangian Relaxation-based Algorithm to Solve a Home Health Care Routing Problem," *International Journal of Engineering,* vol. 31, pp. 1734-1740, 2018.

[22] J. F. BENDER, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik,* pp. 238-252, 1962.

[23] G. J.N. Hooker, "Logic-based Benders decomposition," *Springer-Verlag,* p. 33–60, 2003.

[24] C. Luong, "An examination of Benders' decomposition approaches in large-scale healthcare optimization problems," *University of Toronto Master thesis,* 2015.

[25] B. N. e. al, "A novel and efficient exact technique for integrated staffing, assignment, routing, and scheduling of home care services under uncertainty," *Omega,* vol. 116, 2023.

[26] W. J. a. C. Blume, "Pareto Optimization or Cascaded Weighted Sum: A Comparison of Concepts," *algorithms,* pp. 166-185, 2014.

[27] R. T. M. ·. J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Springer-Verlag,* p. 853–862, 2009.

[28] I. e. a. Muter, "Benders Decomposition and Column-and-Row Generation for Solving Large-Scale Linear Programs with Column-Dependent-Rows.," *European Journal of Operational Research,* pp. 29-45, 2018.