A Genetic Algorithm approach to a Workforce Planning problem

Applied to Erasmus MC

by

A. Gamboa

to obtain the degree of Master of Science in Applied Mathematics
at the Delft University of Technology,
to be defended publicly on Thursday November 25, 2021 at 15:00.

Student number:	5027063	
Project duration:	March, 2021 – November, 2021	
Thesis committee:	Dr. ir. J. H. Weber,	TU Delft, chair
	Dr. ir. J. T. van Essen,	TU Delft, supervisor
	Dr. S. Hinrichs-Krapels,	TU Delft

TU Delft

Preface

This thesis finalizes the work I did in collaboration with the Erasmus University Medical Center (Erasmus MC). During this project, I was constantly amazed by the reach and impact of mathematics on real-life problems. I learnt about the struggles and challenges the nursing personnel of Erasmus MC faced during the COVID-19 pandemic. I had the incredible opportunity to work with the Capacity Management team of the hospital to aim to avoid a future shortage of nurses as the one Erasmus MC faced during these difficult times.

I want to express my gratitude to Cobus van Wyk for considering me for this project; the knowledge and experience I got from working with you are priceless. Furthermore, I am more than grateful for the colleagues that provided the required data for this thesis; Anne Krajnc, Pascale Thielemans, Astrid van der Horst, and Madelon de Bree. Thank you all for the time you spent with me answering all the questions I had about the nursing personnel within Erasmus MC. In addition, I would like to express my gratitude to the Capacity Management team of Erasmus MC; thank you for making me feel part of the team and for the encouragement throughout my thesis.

Moreover, I feel so lucky to have had Theresia van Essen as my supervisor; I cannot thank you enough for your patience and guidance during this thesis. Especially during these pandemic times, I truly appreciate that you cared about my project and that you cared for me and my mental health.

Lastly, I want to show my sincere appreciation towards the people that made my master's at TU Delft special. To Lisa, our study sessions and our lunch breaks were precious to me; thank you for going through this journey with me. To Jade, even in the distance and with different time zones, you were always there for me; thank you for being part of my life. And to my parents, for their unconditional love and support, this master's could not have been possible without you.

A. Gamboa
Delft, November 2021

Contents

1 Introduction

Erasmus University Medical Centre (Erasmus MC), founded in 2002, is an organisation that provides a full spectrum of clinical services, including those supplied by Erasmus MC Sophia Children's Hospital and Erasmus MC Cancer Institute, which fall under the Erasmus MC umbrella. In addition, the core tasks Erasmus MC focuses on are: providing patient care, teaching, training, and conducting research.

Furthermore, Erasmus MC is affiliated with Erasmus University and is the home of its faculty of medicine. It ranks number one in the top European institutions in clinical medicine and it is the largest scientific University Medical Centre in Europe.
Like most hospitals, Erasmus MC faced an evident nursing personnel shortage in the Intensive Care Unit (ICU) when the first wave of COVID-19 arrived. However, even though the pandemic made this workforce shortage problem evident, said problem has been present even before the current pandemic.

Moreover, currently in Erasmus MC, there is sub optimal lines of communication between the rostering departments, which are in charge of nursing scheduling, and there is no view on the needed training between departments, which results in a problem when there is a shortage of personnel in one or more departments. These factors make it harder to have a more efficient scheduling of nurses and a more flexible workforce. Hence, this work is a way to link the hospital's departments and to create a possible bridge of communication in the future for better planning.

The management team of Erasmus MC would like for wards to work together by implementing cross-training and transferring nurses between wards, so in the future the hospital would be better prepared for fluctuating and irregular demand. The motivation for this project is to show that it is possible to provide the required health care while preparing a more flexible workforce. The overall objectives for this thesis are:

→ To propose an efficient use of nursing personnel.

→ To implement cross-training in an intelligent way for a more flexible workforce.

→ To provide the required health care patients need.

→ To encourage priority training.

→ To minimize overall costs; these costs include hiring external nurses and the time nurses spent training on the job.

In order to apply this project to Erasmus MC, it is important to have a clear understanding of the nursing personnel's structure within the hospital. It is required to understand how training works in each department, to know the structural cooperation that takes place within Erasmus MC, and other technical aspects. Examples of these features within the hospital are:

- There are two types of training; advanced training, which takes place outside the workplace and the dates of this training are fixed. And on the other hand, there is training on the job, which takes place within the hospital and there are no restrictions on the dates when this training take place.

- The type of advanced training each nurse can take. For example, nurses from the Dijkzigt Movement ward can take the Traumatology specialty advanced training, but do not take the Oncology specialty one.

- The duration of training on the job among wards; for example, in the Oncology ward, it takes up to six months to train a nurse to provide full-time health care in said ward, while in the Gynecology ward it takes up to one month.

- The type of contract nurses have, which states how many hours a nurse works per year, and provides the number of shifts nurses can work per week.

To fulfil the project's objectives, a model is created such that: it takes into account available nurses and their respective skill levels, possible advanced training each nurse can take to upgrade her/his skills along with the cost and duration of each training, the possibility of cross-training within departments, the hospital's demand, seen as the amount of health care patients need (in hours) during the fixed time horizon, and the skills for which priority training is aimed to be maximized during said time horizon. The patients are categorized based on the set of skills they need from nurses, so instead of referring to each patient specifically, they will be referred to as patient types.

The model will provide schedules on a daily basis, showing what each nurse does along the time horizon and in which ward; each nurse either works or has a rest day, and if a nurse works, he/she either attends advanced training, provides full-time health care, or is training on the job. In addition, the model gives the hours each nurse will spend with each type of patient for each day in the time horizon. All of these assignments are done while ensuring all patients are getting the required health care and the capacity each nurse has available is respected.

This thesis is composed as follows: Chapter 2 displays the relevant literature for this project, showing with a brief summary the work that has been done in related projects and the relevance this thesis has. In Chapter 3, the model is introduced in a general way, and the mathematical model is presented afterwards. Chapter 4 explains the solution method that was implemented to solve the model. Next, Chapter 5 discusses the parameters provided by the hospital in order to implement the model previously defined. This chapter also contains the required data pre-processing that is done (if applicable). Chapter 6 displays and discusses the results of these implementations, and Chapter 7 reflects on the whole project and provides further recommendations. Finally, conclusions can be found in Chapter 8.

2
Literature Review

This chapter discusses the relevant literature for this project. For each paper, a brief summary is given pointing out key factors for this thesis.
Firstly, the work of De Bruecker et al. [2015] was the main source of literature for this thesis. The authors of this article provide a rigorous classification of the literature that deals with workforce planning incorporating skills. Some of the categories used are the following;

- Literature using hierarchical and categorical skill types.
- Allowing substitution or not when dealing with hierarchical skills.
- Literature that implements training.
- Projects that manage learning and forgetting skills.
- Application areas.

This article provided most of the literature that was used for this project, and the preceding reviewed papers that are not stated otherwise, were provided by this article. Section 2.1 provides the literature used as a background to build the mathematical model of the thesis, and Section 2.2 contains papers that provided potential approaches to solve it.

2.1. Mathematical Modelling

This section displays relevant literature related to the problem this thesis aims to solve. Subsection 2.1.1 shows literature that only considers workers' skills, and Subsection 2.1.2 presents papers that incorporate the possibility of training to gain and/or upgrade skills.

2.1.1. Literature focused only on Skills

First of all, Li and Li [2000] present a multiple objective goal programming (GP) method for a multi-skilled personnel planning problem. This problem aims to determine the staff needed for the planning period within budgetary constraints. The authors apply this model to real data from an AIDS prevention clinic in China. It takes into consideration task substitution, regular demand and irregular demand (non-scheduled patients), overtime, and hiring and firing staff.

Next, Cai and Li [2000] consider the staff scheduling problem with mixed skills. The problem considers two types of skills and a workforce where each employee has at least one of these skills. The model aims to assign each worker to a weekly schedule fulfilling each task's demanded skill. In this paper, each job requires either skill 1 or skill 2 to be completed, so there are no jobs that require both skills. The authors consider three different objective functions ranked by priority: minimize overall cost, maximize staff surplus, and minimize the variation of staff surplus. They propose a new Genetic Algorithm for working with these three different criteria.

Furthermore, Wallace and Whitt [2005] implement a local search algorithm to help improve management decisions in call centers. Usually, call centers handle different types of calls where an agent requires a certain skill to handle each type of call. This work aims to find the optimal number of agents and the optimal number of skills said agents should have in order to efficiently cover the expected calls.

Another paper within the health care category is provided by Bard and Purnomo [2005]. The authors work with the nurse scheduling problem taking into account nurses' preferences and nurses' skills. In this paper, hierarchical skills are considered and they are used to implement downgrading among nurses; in other words, a high skill level nurse can be assigned assigned to a shift that requires a low skill level nurse. The authors perform a column generation approach along with heuristics to recognize good rosters. Even though the presented model considers nurses' skills, it does not consider training to upgrade nurses' skills.

Sayin and Karabati [2007] aim to assign cross-trained workers to different departments. The authors do this by applying a two phase approach where the first phase consists of maximizing departmental utility and the second part consists of maximizing skill improvement among the workforce. The paper considers each worker's skill level at each department, and a hyperbolic learning curve is used to asses the benefits of each assignment taking into account the time a worker has been in each department. It is a general approach since the model is not applied to a specific real case.

Furthermore, the work of Tiwari et al. [2009] is based on the multi-mode, resource-constrained, project scheduling problem (MRCPSP). This problem works with similar skill level resources to perform the required jobs aiming to minimize the total project makespan. However, the authors implement a variant of this problem allowing to work with different levels of skills. The article states that with this approach, there is the possibility for a low-level worker starting a project (or a part of it) and a high-level worker can then complete the project to attain the expected quality level. From the results of the implementation of this problem, further analysis is made to identify cross-training opportunities.

2.1.2. Literature that incorporates Training

The work provided by DePuy et al. [2006] is not in the literature considered by De Bruecker et al. [2015], the former deals with the problem of assigning workers to tasks in a company based on the worker's skill level. The authors develop two models; the first one is intended to be applied if there is a lack of time for workers to raise their skill level, and the aim is to make the assignment between workers and tasks, while minimizing the gap of each worker's skill level with the skill level the task requires. The second model is called the skills management problem and introduces the possibility for workers to receive training in order to raise their skill level. This model decides the workers that will go under training, and makes the assignment of the tasks with the workers taking into account the worker's updated skill level. The objective function of this model is to minimize the overall training costs. To the best of our knowledge, this is the first paper that addresses the skills management problem, which suited this project the best.

Moreover, Fowler et al. [2008] aim to make different staff decisions like firing, cross-training or hiring employees, while minimizing the overall costs of these decisions and fulfilling the demanded work. The authors implement the use of skills and the general cognitive ability (GCA) of their employees to make these staff decisions. GCA is defined as the ability to learn or process information. The model of this paper allows skills not being satisfied. They implement two linear programming based heuristics, and a genetic algorithm to compare the performance of the heuristics.

Projects related with the one of DePuy et al. [2006] are the ones by Grieshaber [2009] and Jackson [2009]; these are theses that were supervised by G. Depuy. Grieshaber [2009] aims to improve existing algorithms to upgrade efficiency in large data sets for the skills management problem. The author makes use of Genetic Algorithms incorporating preexisting Greedy Algorithms to compare results, running time, and overall efficiency of the different algorithms implemented. On the other hand, Jackson [2009] implements three different heuristic techniques to the skills management problem and aims to compare these by executing them with randomly generated data sets. The heuristic techniques used are: a greedy assignment algorithm, Meta-RaPS greedy heuristic, and Meta-RaPS shortest augmenting path.

The work of Huang et al. [2009] presents a simulator called *SimMan* that assesses different managerial decisions that affect overall workforce planning; such as whether to allow cross-training or not, and if a delay in the starting time of a task can be permitted or not. This evaluation then is used to compare these managerial decisions and their effects in the solutions provided. They present one planning model that considers skill-based demand and cross-training of the workforce to fulfill the fluctuating demand of a company. It is important to notice that the skills used are not hierarchical. This model deals also with hiring personnel, firing personnel, and outsourcing when the demand cannot be met. Some results of the implementation of the simulator *SimMan* showed that when a company allows cross-training, there are significant savings since less personnel is needed, and outsourcing decreases when the start of a task can be delayed.

Additionally, Smet et al. [2014] propose a generic model to the nurse rostering problem. This work addresses the complexity of real-world nurse rostering problems and the lack of practicability in the available academic models for these. Consequently, the authors introduce basic concepts that are usually used in real-world nurse rostering problems and the different scenarios that can be presented in such concepts. For example, when nurses' skill types are introduced, it is stated that there can be three different scenarios; either all employees have the same skill type (which is unusual in real-world problems), each employee has at most one skill type, or employees can have more than one skill type. This paper implements a hyper-heuristic approach to solve the nurse rostering problem applied to real-case scenarios from Belgian hospitals, comparing its performance to the approach of an adaptive large neighbourhood search.

2.1.3. Conclusion

Even though some work has been done in the workforce planning area, the body of literature considered does not support approaches for the proposed problem in this project. As already shown, most papers that deal with multi-skilled workforce do not consider training to upgrade a worker's skill level, and the ones that do, do not have a breakdown of how the training will be implemented in the future; they just decide whether a worker will receive training in a certain skill or not.

Furthermore, papers that are in the health care category mostly focus on the nurse rostering problem; one of the exceptions is the work of Li and Li [2000]. However, this article considers staff skills only for possible substitution among the workforce and it does not consider training. The work that was the closest to this project is the one by DePuy et al. [2006], specifically the model for the skills management problem. Nevertheless, this is not exactly what the model of this project aims for; this thesis aims to provide the exact periods when nurses will receive training throughout the time horizon, while DePuy et al. [2006] do not work with a time horizon, nor a schedule for the planned trainings. However, this paper is used as the base for building the mathematical model of this thesis.

2.2. Solution Methods

In this section, literature that implements possible solution methods for the model implemented in this thesis is displayed. This section will mainly focus on the techniques used to solve different workforce planning problems and at the end it will be discussed which method will be applied in this thesis.

Authors Cai and Li [2000] solve a scheduling problem that considers two types of jobs and workers with at most two different skills; and the type of job a worker is able to do depends on the skill(s) he/she has. They solve this problem using a Genetic Algorithm (GA); the proposed GA implements a ranking scheme in the parent selection phase of the algorithm. It executes the uniform crossover operator, and it uses a heuristic to solve infeasibility after crossover operators.

Moreover, Bard and Purnomo [2005] work with the nurse rostering problem taking into account nursing personnel's preferences. The authors not only focus on getting efficient schedules, they aim to provide quality rosters for nurses taking into account nurses' skills. The problem presented considers the employment of outside nurses and the excess number of nurses used for each period.
This work solves the proposed problem with two different approaches using a column

generation-based technique; the approaches differ on how the algorithm meets the demand of nurses for each skill.

Research by Fowler et al. [2008] implements two linear programming (LP) based heuristics and a genetic algorithm (GA) approach to a workforce planning problem. The problem tackled in this paper considers periods throughout a time horizon, and for each period, decisions like hiring, training, and firing are made.
The LP based heuristics solve the relaxation of the problem, then they either round up or down the variables that are meant to be integer. The difference between these two LP based heuristics is that one solves the relaxation of a problem for each period, and the other only solves the relaxation of the problem once. At the end, the authors implement the GA only to evaluate the performance of these two heuristics; it is important to notice that the implemented GA outperformed the LP based heuristics.

The work done by Grieshaber [2009] and Jackson [2009] are theses that were done under the supervision of G. Depuy. Both papers implemented heuristics to solve the skills management problem introduced by DePuy et al. [2006].
Grieshaber [2009] implements a genetic algorithm approach while Jackson [2009] used a Shortest Augmenting Path (SAP) algorithm. The skills management problem is a special case of the generalized assignment problem, and since both of these approaches are commonly used to solve assignment problems, it makes sense for the authors to have implemented them.

Furthermore, Bai et al. [2010] implement a Genetic Algorithm (GA) combined with a Simulated Annealing Hyper-Heuristic (SAHH) approach to solve a nurse rostering problem. The authors state that genetic algorithms can be improved by combining them with local search procedures, specifically, combining the GA with the SAHH approach has shown stunning results for difficult problems.
Hyper-heuristics are high-level techniques that deal with a pool of heuristics in order to widen the solution space. Particularly for this paper, the SAHH approach uniformly chooses one among nine neighbourhoods to get a new solution for each individual of the population, and then apply the crossover and mutation operators to the obtained population.

Additionally, Ho and Leung [2010] use a manpower scheduling problem to model airline catering operations. The model used in this research considers time windows and job-skills constraints; it contemplates workers with at most two different skills, and it aims to form assignments between teams of two persons with flights seen as jobs.
Authors in this paper approach the problem with two heuristics: Tabu Search and Simulated Annealing, showing that the Tabu Search approach outperformed the Simulated Annealing one.

Smet et al. [2014] implement a hyper-heuristic approach to solve a nurse rostering problem. This approach considers two main features; a selection phase and an acceptance criterion. It selects one low level perturbation heuristic at a time, a neighbour from the current solution is obtained and then it is assessed if the new solution is accepted or not.
Working with several neighbourhoods enables a broader search in the solution space. Ex-

amples of the neighbourhoods used in this paper are: assigning a shift to a nurse, deleting an existing shift from a nurse, and general assignment of shifts change.

Wang et al. [2021] present two mixed-integer programming formulations that model single and multiple period operations (like hiring and cross-training), motivated by how seasonal businesses work. Seasonal businesses, like agricultural businesses, hire workers depending on external factors like environmental conditions and market prices.
This paper solves these two models through a Tabu Search algorithm using k-OPT strategies, and to compare the solution quality of this approach, they solve several instances for both models with Gurobi solver and with this technique. An k-OPT algorithm is a local search heuristic widely used for the Traveling Salesman Problem.

The literature included in this section shows possible solution methods that can be implemented in this thesis. Most papers were provided by De Bruecker et al. [2015], which gives a classification on the literature that deals with workforce planning based on the solution methods they implemented.
De Bruecker et al. [2015] showed that among the heuristics used to solve a workforce planning problem, the Genetic Algorithm (GA) approach is widely used and has shown to perform quite well to solve assignment problems and workforce planning problems. The displayed papers that used a GA either improved the approach by combining it with other techniques, or used it as a baseline to compare other heuristics.

3
Problem Formulation

In this chapter, the model is thoroughly explained. A general description is discussed first and the mathematical model is introduced after. In addition, the parts that were similarly taken from the work done by DePuy et al. [2006] will be pointed out throughout the explanation of the mathematical model.

3.1. Problem Description

The model considers three main factors: nursing personnel, nurse's skills, and the hospital's demand to fulfill. Given a set of nurses and a set of skills, each nurse has a subset of skills (as graphically exemplified in Figure 3.1). The nurse's acquired skills can depend on several factors, such as prior education, work experience and/or certifications obtained.

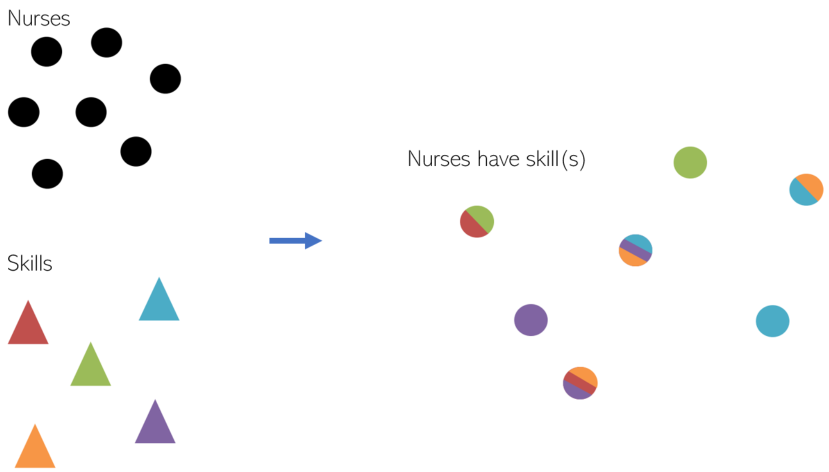


Figure 3.1: Graphical example of a set of nurses, a set of skills, and how each nurse has a specific subset of skills.

Furthermore, the model considers the possibility for nurses to learn and/or upgrade their skills. In other words, it is possible for nurses to receive training in order to acquire new skills, as illustrated in Figure 3.2.

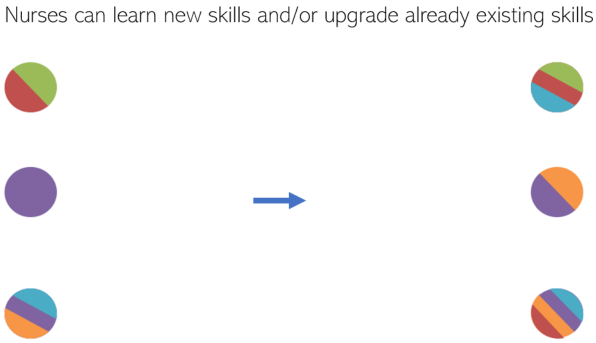


Figure 3.2: Graphical example of how nurses can acquire new skills.

Specifically for this thesis, there are two types of training. One kind of training is training on the job; the aim of this type of training is to prepare nurses so they are able to provide health care in a specific ward. It is given within the hospital and there are no fixed dates for this training.

However, this training begins when a nurse starts working in a ward, and the duration of the training depends on the prior experience of each nurse. For example, in the Gynecology ward, nurses with no work experience take up to twelve months to fully provide health care in said ward, whereas experienced nurses may take four weeks to do so.

Moreover, when a nurse is training on the job there are two components to be considered: the amount of supervision he/she requires and the quantity of health care he/she can provide on his/her own. When training on the job starts, nurses need full-time supervision, hence they cannot provide any health care by their own. Throughout the training, the amount of supervision nurses need is considered to be decreasing linearly until no supervision is necessary. Therefore, the quantity of health care nurses can provide by their own is increasing linearly, until they can provide full-time health care in their ward.

A graphical example of these two factors during training on the job can be seen in Figure 3.3.

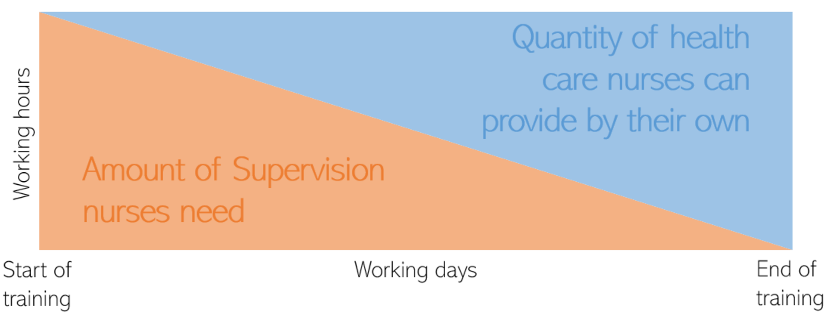


Figure 3.3: Graphical example of the amount of supervision needed from nurses versus the quantity of health care they can provide by their own throughout training on the job.

On the other hand, there is advanced training; this training takes place outside the hospital and it has fixed dates. The hospital decides which nurses are ready to take this training,

and the objective of this training is for nurses to gain specialized skills, so that they are able to provide specific health care in their own ward or to get the skills they lack to be able to provide health care in another ward.
Nurses that have the skills to provide health care on multiple wards are considered to be flexible. Having a flexible workforce in the hospital gives the possibility to fulfill irregular demand without any personnel shortage.

The hospital's demand that is aimed to fulfill is seen as the amount of health care provision (in hours) patients require, where a patient is represented as a time interval that starts the moment the patient is admitted to the hospital and ends at the moment the patient gets discharged. During their stay, patients require nurses with specific skills to provide them the adequate health care, as graphically shown in Figure 3.4.
It is important to notice that for this project specifically, the model will work with historical data of the patients. This means that the wards where patients get treated are fixed.

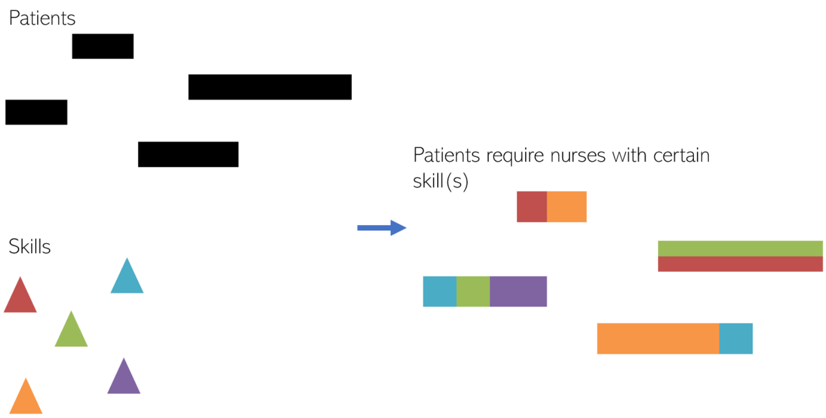


Figure 3.4: Graphical example of a set of patients, a set of skills, and how each patient has specific skill needs.

The model works with periods (days) throughout a fixed time horizon, and certain decisions must be made, such as:

- The nurses that should receive training, which training they should get and when
- If there is need to hire external nurses to fulfill the hospital's demand
- If a nurse works/receives training/rests in a specific day
- The nurses that will be included in the flexible workforce
- For the nurses in the flexible workforce, which ward they should work in for each period
- For each week in the time horizon, how many days each nurse works; this decision must be made taking into account the contractual obligations nurses have with the hospital.

Given the features discussed above, what the model aims to provide is a way to fulfill the hospital's demand while building flexible nursing personnel by implementing training among nurses, doing it in an efficient way so the overall training costs are minimized and priority training for specific skills is applied.
A graphical example of the problem can be seen in Figure 3.5.

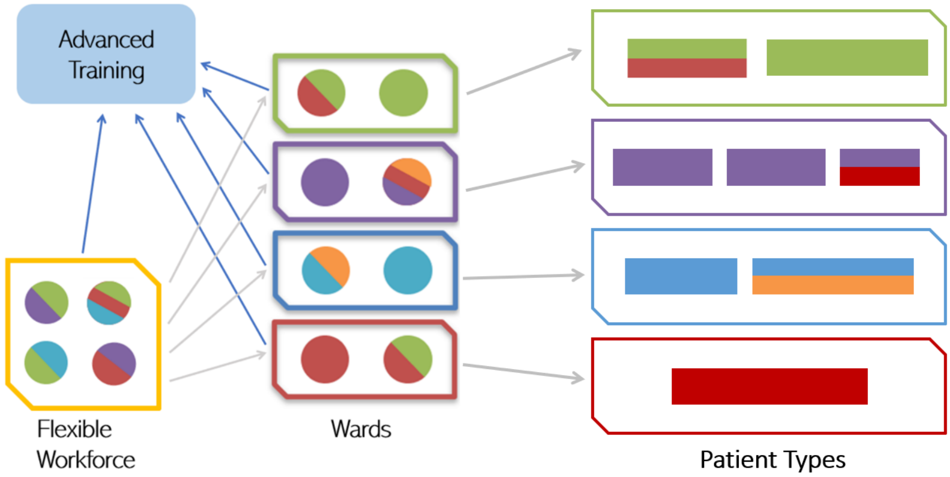


Figure 3.5: Graphical example of the decisions that the model will make for each period in the time horizon. For nurses that are flexible, it is decided which ward they work in to fulfill the required health care by patients in said ward, or if they take advanced training to gain new skills. For non-flexible workforce, it is decided whether they work to meet the needed health care in their own ward, or if they take advance training to gain new skills.

3.2. Mathematical Model

To bring this problem into a mathematical model, there are several things that need to be taken into account and quantified; first of all, the set of skills and nurses are given by sets K and N, respectively, the time horizon for which the model is applied to is given in days by the set T, and let I be the set of weeks the model takes into account. Also, S_{nk} is introduced to keep track of the skill level each nurse $n \in N$ currently has in skill $k \in K$, and let a_k be the highest level a nurse can have in skill $k \in K$.

Connected to this, let N_k be the set of nurses that have skill $k \in K$ at its highest level (i.e. $N_k = \{n \in N | S_{nk} = a_k\}$), in other words, N_k contains the nurses that do not need further training in skill $k \in K$. In addition, let D_n be the number of days nurse $n \in N$ must have worked at the end of the time horizon T, and let δ_n^{min}, δ_n^{max} be the minimum and maximum number of days nurse $n \in N$ can work weekly, respectively.

Now, to introduce the parameters linked to the hospital's demand, which is seen as patients, it is important to notice that a set of patients is not considered per se; instead, each patient is categorized based on the health care he/she needs in terms of nurses' skills, so each patient is labeled as a patient type, and set J contains all patient types the model considers. Moreover, H_{jt} gives the hours each patient type $j \in J$ needs health care from a nurse on day $t \in T$, and R_{jk} gives the required level of skill $k \in K$ a nurse must have to provide health care to patient type $j \in J$.

Moreover, let W be the set of wards in the hospital, let $w_n \in W$ be the ward of nurse $n \in N$, and let w_j be the ward where patient type $j \in J$ belongs to.

Since this project considers the possibility for nurses to get new skills and/or to upgrade their current skills, parameters linked to training are needed. Regarding the advanced training, i.e. the training that is fixed, let $\Theta_{km} \subseteq T$ contain the days on which advanced training takes place to improve skill k to level m, and let θ_{km} be the last day of advanced training to improve skill $k \in K$ to level m. If this day is out of the time horizon T, θ_{km} is set to be the last day of the time horizon T. Furthermore, the binary parameter τ_{nkm} is introduced, which is 1 if nurse $n \in N$ is a prospect to take the advanced training to upgrade skill $k \in K$ from level S_{nk} to level m, and 0 otherwise.

On the other hand, for the training on the job: let e_{nw} be the number of days it takes nurse $n \in N$ to complete training in ward $w \in W$. It is important to notice that a nurse can take a training only once, in other words: advanced training to improve skill $k \in K$ can be taken only once, and training on the job to be able to provide health care in ward $w \in W$ can be taken only once.

The parameter e_{nw} can depend in both the nurse's experience and the ward the nurse comes from. In addition, training on the job requires the supervision of an experienced nurse; the nurse that receives the training starts needing full-time supervision from an experienced nurse, and as the training progresses, the needed supervision decreases until there is no supervision needed at all and the trained nurse can provide full health care in the respective ward. This means that while a nurse is under supervision he/she is unable to provide full-time health care in the ward where he/she is training, hence the parameter $f_{nw\beta}$ gives the hours of health care nurse $n \in N$ can provide daily in ward $w \in W$ after β days of training on the job in said ward.

Finally, Erasmus MC aims to have more flexible nursing personnel, and to achieve this, the hospital wants to encourage training in certain skills. Therefore, $P \subseteq K$ is defined as the set of skills for which priority training will be implemented, and let B_{nkm} be the benefit of training nurse $n \in N$ to upgrade skill $k \in K$ from level S_{nk} to level m. In addition, since the aim is to minimize the costs of hiring external nurses and the costs spent on training on the job, let E_j be the hourly cost for hiring an external nurse to provide health care for patient type $j \in J$ and let C_{nw} be the daily cost it takes for nurse $n \in N$ to train on the job in ward $w \in W$.
It is important to notice that the definition for parameters S_{nk} and R_{jk} is similar to the one provided by DePuy et al. [2006].
All the parameters introduced above are displayed in Table 3.1.

Variables

For each nurse $n \in N$, for each skill $k \in K$, and for each skill level m with $S_{nk} < m \leq a_k$, the binary variables V_{nkm} are defined. These variables take value 1 if nurse $n \in N$ takes advanced training to upgrade skill $k \in K$ from level S_{nk} to level m, and 0 otherwise. Variables

V_{nkm} are similarly defined as in the skills management problem provided by DePuy et al. [2006].

In addition, for each nurse $n \in N$, for each ward $w \in W$, and for each day $t \in T$, the binary variables Y_{nwt} take value 1 if nurse $n \in N$ works and is able to provide full-time health care in ward $w \in W$ on day $t \in T$, and 0 otherwise.

Variables X_{njt} give the number of hours nurse $n \in N$ provides health care for patient type $j \in J$ on day $t \in T$.

Furthermore, variables U_{jt} represent the hours of health care that the nursing personnel from the hospital is not able to fulfill for patient type $j \in J$ at day $t \in T$. These variables provide the daily number of hours of health care the hospital must fulfill with external workforce, like temporary nurses, for example.

Moreover, binary variables Z_{nwt} take value 1 if nurse $n \in N$ works and receives training on the job in ward $w \in W$ on day $t \in T$, and 0 otherwise.

Also, binary variables $Q_{nw\beta t}$ take value of 1 if nurse $n \in N$ has trained in ward $w \in W$ for β days by day $t \in T$, and 0 otherwise.

Lastly, binary variables O_n take value of 1 if nurse $n \in N$ is flexible and 0 otherwise.
All the variables introduced above are shown in Table 3.2

Auxiliary Variables

Binary auxiliary variables W_{njt} are defined so they take the value of 1 if nurse $n \in N$ provides health care to patient type $j \in J$ on day $t \in T$, and 0 otherwise. In other words, W_{njt} equals 1 if $X_{njt} > 0$, and 0 otherwise.

In addition, binary auxiliary variables λ^1_{nwt}, λ^2_{nwt} are introduced for every nurse $n \in N$, for every ward $w \in W$, and for every day $t \in T$. These are defined so that they both take the value of 1 if nurse $n \in N$ has started training on the job in ward $w \in W$, but has not finished it by day $t \in T$. In other words, λ^1_{nwt} takes value of 1 if $\sum_{u=1}^{t} Z_{nwu} \geq 1$, and 0 otherwise. And λ^2_{nwt} takes value of 1 if $\sum_{u=1}^{t} Z_{nwu} < e_{nw}$, and 0 otherwise.
All the auxiliary variables introduced above are shown in Table 3.3

Constraints

Constraints (3.1) state the relation between the hours of health care nurse $n \in N$ provides for patient type $j \in J$ in ward $w \in W$ on day $t \in T$ and the variables that define whether nurse $n \in N$ works in ward $w \in W$ on day $t \in T$, or not.

$$X_{njt} \leq 8Y_{nwt}, \qquad \forall w \in W, \forall j \in J \text{ such that } w_j = w, \forall n \in N, \forall t \in T. \tag{3.1}$$

Moreover, Constraints (3.2) make sure that on day $t \in T$ nurse $n \in N$ either works, receives training, or neither.

$$\sum_{w \in W} Y_{nwt} + \sum_{w \in W} Z_{nwt} + \sum_{k \in K} \sum_{m > S_{nk}}^{a_k} \chi_{\Theta_{km}}(t) V_{nkm} \leq 1, \qquad \forall t \in T, \forall n \in N. \tag{3.2}$$

Where $\chi_{\Theta_{kS_{nk}m}}$ is the indicator function for set $\Theta_{kS_{nk}m}$. Additionally, Constraints (3.3) respect the number of days nurse $n \in N$ works or trains, throughout the time horizon, according to the nurse's contract with the hospital.

$$\sum_{t \in T} \sum_{w \in W} Y_{nwt} + \sum_{t \in T} \sum_{w \in W} Z_{nwt} + \sum_{k \in K} \sum_{m > S_{nk}}^{a_k} V_{nkm} |\Theta_{km}| = D_n, \qquad \forall n \in N. \tag{3.3}$$

Moreover, Constraints (3.4) and (3.5) make sure that the minimum and maximum number of days nurse $n \in N$ works or trains weekly is respected, respectively. These constraints take into account the seven days for each week the model considers, and for each day, the constraints track whether a nurse worked or took training.

Constraints (3.4) and (3.5) follow the same idea as DePuy et al. [2006], when the skills management problem is introduced and the constraints of capacity for each worker are defined.

$$\begin{aligned}\delta_n^{min} \leq \sum_{w \in W} \sum_{j=1}^{7} Y_{nw(7(i-1)+j)} &+ \sum_{w \in W} \sum_{j=1}^{7} Z_{nw(7(i-1)+j)} \\ &+ \sum_{k \in K} \sum_{m > S_{nk}}^{a_k} \sum_{j=1}^{7} \chi_{\Theta_{km}}(7(i-1)+j) V_{nkm}, \qquad \forall i \in I, \forall n \in N.\end{aligned} \tag{3.4}$$

$$\begin{aligned}\sum_{w \in W} \sum_{j=1}^{7} Y_{nw(7(i-1)+j)} &+ \sum_{w \in W} \sum_{j=1}^{7} Z_{nw(7(i-1)+j)} \\ &+ \sum_{k \in K} \sum_{m > S_{nk}}^{a_k} \sum_{j=1}^{7} \chi_{\Theta_{km}}(7(i-1)+j) V_{nkm} \leq \delta_n^{max}, \qquad \forall i \in I, \forall n \in N.\end{aligned} \tag{3.5}$$

Furthermore, Constraints (3.6) put an upper bound on training on the job. These constraints make training on the job for ward $w \in W$ take place once for nurse $n \in N$ throughout the time horizon.

$$\sum_{t \in T} Z_{nwt} \leq e_{nw}, \qquad \forall n \in N, \forall w \in W. \tag{3.6}$$

Constraints (3.7) condition the relation between variables $Q_{nw\beta t}$ and Z_{nwt} for every nurse $n \in N$, ward $w \in W$, and day $t \in T$.

$$\sum_{\beta=1}^{\min\{t, e_{nw}\}} \beta Q_{nw\beta t} = \sum_{u=1}^{t} Z_{nwu}, \qquad \forall n \in N, \forall w \in W, \forall t \in T. \tag{3.7}$$

In addition, Constraints (3.8) put an upper bound on the variables $Q_{nw\beta t}$ for every nurse $n \in N$, ward $w \in W$, and day $t \in T$.

$$\sum_{\beta=1}^{\min\{t,e_{nw}\}} Q_{nw\beta t} \leq 1, \qquad \forall n \in N, \forall w \in W, \forall t \in T. \tag{3.8}$$

Constraints (3.9) ensure that nurse $n \in N$ works full-time in ward $w \in W \setminus \{w_n\}$ only after finishing training on the job in said ward. For these constraints, a value M_1 is chosen big enough such that $M_1 \geq \sum_{u \leq t} Z_{nwu} + 1 - e_{nw}$, for every nurse $n \in N$, every ward $w \in W$, and every day $t \in T$.

$$e_{nw} - \sum_{u=1}^{t} Z_{nwu} \leq M_1\Big(1 - Y_{nwt}\Big), \qquad \forall n \in N, \forall t \in T, \forall w \in W \setminus \{w_n\}. \tag{3.9}$$

Furthermore, Constraints (3.10) force the definition of auxiliary variables λ^1_{nwt}, Constraints (3.11) force the definition of λ^2_{nwt}, and Constraints (3.12) guarantee that training on a job goes uninterrupted, i.e., once nurse $n \in N$ starts training on the job in ward $w \in W$, then said nurse cannot provide any health care in any other ward until he/she finishes his/her current training on the job. For these constraints, a value M_2 is chosen such that $M_2 > e_{nw}$ for every nurse $n \in N$, and every ward $w \in W$.

$$\sum_{u=1}^{t} Z_{nwu} < 1 + M_2\lambda^1_{nwt}, \qquad \forall n \in N, \forall t \in T, \forall w \in W. \tag{3.10}$$

$$\sum_{u=1}^{t} Z_{nwu} \geq e_{nw} - M_2\lambda^2_{nwt}, \qquad \forall n \in N, \forall t \in T, \forall w \in W. \tag{3.11}$$

$$\sum_{w' \in W} Y_{nw't} \leq 2 - \lambda^1_{nwt} - \lambda^2_{nwt}, \qquad \forall n \in N, \forall t \in T, \forall w \in W. \tag{3.12}$$

Constraints (3.13) ensure that nurse $n \in N$ can do advanced training for skill $k \in K$ at most once throughout the time horizon.

$$\sum_{m > S_{nk}}^{a_k} V_{nkm} \leq 1, \qquad \forall k \in K, \forall n \in N. \tag{3.13}$$

Also, Constraints (3.14) ensure only the prospects for advanced training can take the advanced training.

$$V_{nkm} \leq \tau_{nkm}, \qquad \forall n \in N, \forall k \in K, \forall m > S_{nk}. \tag{3.14}$$

Constraints (3.15) make sure each patient type $j \in J$ receives the respective hours of health care needed on day $t \in T$.

$$\sum_{n \in N} \sum_{\beta=1}^{\min\{t,e_{nw}\}} f_{nw_j\beta} Q_{nw_j\beta t} + \sum_{n \in N} X_{njt} + U_{jt} = H_{jt}, \qquad \forall j \in J, \forall t \in T. \tag{3.15}$$

Moreover, Constraints (3.16) force the definition of auxiliary variables W_{njt}; they will take value of 1 if nurse $n \in N$ provides health care to patient type $j \in J$ on day $t \in T$, and 0 otherwise.

$$X_{njt} \le 8W_{njt}, \qquad \forall j \in J, \forall n \in N, \forall t \in T. \tag{3.16}$$

In addition, Constraints (3.17) and (3.18) make sure that patient type $j \in J$ gets the required level of health care in each skill $k \in K$. Constraints (3.17) consider the periods of the time horizon when advanced training has not ended yet, and (3.18) consider the periods when advanced training has ended. Constraints (3.17) and (3.18) follow the same idea as DePuy et al. [2006], when the skills management problem is introduced and the constraints that ensure each task gets performed by a worker with certain skill levels are defined.

$$S_{nk} \ge R_{jk} W_{njt}, \qquad \forall j \in J, \forall k \in K, \forall n \in N \setminus N_k, \forall w \in W, \forall t \le \theta_{km}. \tag{3.17}$$

$$S_{nk}\Big(1 - \sum_{m>S_{nk}}^{a_k} V_{nkm}\Big) + \sum_{m>S_{nk}}^{a_k} mV_{nkm} \ge R_{jk} W_{njt},$$
$$\forall j \in J, \forall k \in K, \forall n \in N \setminus N_k, \forall w \in W, \forall t > \theta_{km}. \tag{3.18}$$

Lastly, Constraints (3.19) keep track of the nurses that are flexible. In other words, if a nurse is able to provide full-time health care in at least two wards (including his/her own ward), then said nurse is considered flexible.

$$\sum_{u=1}^{t} Z_{nwu} - e_{nw} + 1 \le O_n, \qquad \forall n \in N, \forall t \in T, \forall w \in W \setminus \{w_n\}. \tag{3.19}$$

Objective Function

To finalize the mathematical formulation, the Objective Function (3.20) is set to minimize the overall costs of training on the job, to minimize the costs of hiring temporary nurses needed to fulfill the demand that the hospital's workforce cannot fulfill, to minimize the number of flexible nurses, and to maximize the benefit of implementing priority training.

$$\min \sum_{n \in N} \sum_{w \in W} C_{nw} \sum_{t \in T} Z_{nwt} + \sum_{j \in J} E_j \sum_{t \in T} U_{jt} + \sum_{n \in N} O_n - \sum_{n \in N} \sum_{k \in P} \sum_{m>S_{nk}}^{a_k} B_{nkm} V_{nkm} \tag{3.20}$$

K	Set of skills
N	Set of nurses
T	Set of days in time horizon
I	Set of weeks in time horizon
S_{nk}	Skill level of nurse $n \in N$ for skill $k \in K$
a_k	Maximum level a nurse can have in skill $k \in K$
N_k	Set of nurses that have skill $k \in K$ at the highest possible level, $N_k \subseteq N$
D_n	Number of days nurse $n \in N$ must work throughout the time horizon T
δ_n^{min}	Minimum number of days nurse $n \in N$ can work in a week
δ_n^{max}	Maximum number of days nurse $n \in N$ can work in a week
J	Set of patient types
H_{jt}	Hours of health care patient type $j \in J$ needs on day $t \in T$
R_{jk}	Required level of skill $k \in K$ for patient type $j \in J$
W	Set of wards within the hospital
w_n	Ward where nurse $n \in N$ belongs to
w_j	Ward where patient type $j \in J$ belongs to
Θ_{km}	Set of days when advanced training takes place to improve skill k to level m
θ_{km}	Last day of advanced training to improve skill k to level m, if advanced training does not finish within the time horizon, this parameter is set to be the last day in the time horizon T
τ_{nkm}	Binary parameter that is 1 if nurse $n \in N$ is a prospect to take the advanced training to upgrade skill $k \in K$ to level m, and 0 otherwise.
e_{nw}	Number of days it takes nurse $n \in N$ to complete training to provide full-time health care in ward $w \in W$
$f_{nw\beta}$	Hours of health care nurse $n \in N$ can provide daily in ward $w \in W$ after β days of training on the job in ward $w \in W$
P	Set of skills for which priority training will be applied, $P \subseteq K$
C_{nw}	Daily cost it takes for nurse $n \in N$ to train on the job in ward $w \in W$
E_j	Hourly cost for hiring an external nurse to provide health care for patient type $j \in J$
B_{nkm}	Benefit of training nurse $n \in N$, to upgrade skill $k \in K$, from level S_{nk} to level m

Table 3.1: Parameters needed for the mathematical model.

$$V_{nkm} = \begin{cases} 1, \text{ if nurse } n \in N \text{ receives advanced training to improve skill } k \in K \\ \quad \text{from level } S_{nk} \text{ to level } m, \\ 0, \text{ otherwise.} \end{cases}$$

$$Y_{nwt} = \begin{cases} 1, \text{ if nurse } n \in N \text{ works in ward } w \in W \text{ on day } t \in T, \\ 0, \text{ otherwise.} \end{cases}$$

X_{njt} = Hours of health care nurse $n \in N$ provides to patient type $j \in J$ on day $t \in T$.

U_{jt} = Hours of health care the hospital must fulfill for patient type $j \in J$ at day $t \in T$ with external workforce.

$$Z_{nwt} = \begin{cases} 1, \text{ if nurse } n \in N \text{ receives training on the job in ward } w \in W \text{ on day } t \in T, \\ 0, \text{ otherwise.} \end{cases}$$

$$Q_{nw\beta t} = \begin{cases} 1, \text{ if nurse } n \in N \text{ has trained in ward } w \in W, \text{ for } \beta \text{ days by day } t \in T, \\ 0, \text{ otherwise.} \end{cases}$$

$$O_n = \begin{cases} 1, \text{ if nurse } n \in N \text{ is a flexible nurse,} \\ 0, \text{ otherwise.} \end{cases}$$

Table 3.2: Variables needed for the mathematical model.

$$W_{njt} = \begin{cases} 1, \text{ if } X_{njt} > 0 \\ 0, \text{ otherwise.} \end{cases}$$

$$\lambda^1_{nwt} = \begin{cases} 1, \text{ if } \sum_{u=1}^{t} Z_{nwu} \geq 1 \\ 0, \text{ otherwise.} \end{cases}$$

$$\lambda^2_{nwt} = \begin{cases} 1, \text{ if } \sum_{u=1}^{t} Z_{nwu} < e_{nw} \\ 0, \text{ otherwise.} \end{cases}$$

Table 3.3: Auxiliary variables needed for the mathematical model.

4

Genetic Algorithm

The Genetic Algorithm (GA) is an abstraction of biological evolution based on Darwin's theory of natural selection and was developed in the 1960s and 1970s by John Holland and his colleagues. Since then, plenty of variants of the GA have been developed in order to solve a wide range of optimization problems; according to Xin-She [2021], genetic algorithms are known for their ability to deal with complex problems and for outperforming traditional optimization algorithms. The principal definitions of Genetic Algorithms are; a fixed-size set of solutions called a *population*, each solution in the population is called an *individual* or *chromosome*, the components of each individual are called *genes*, and one iteration of creating a new population is called a *generation*.
De Jong [1988] states that Genetic Algorithms consist of three main elements:

- A Darwinian notion of *fitness*, which determines how good a solution is and consequently how much an individual can influence future generations.
- A *reproduction* phase, where the selection of parents for the offspring of the next generation takes place.
- *Genetic operators* that decide how genes will be inherited from the chosen parents. These are known as the crossover and mutation operators.

The Genetic Algorithm starts with a population of a fixed size. Usually, the size of the population that works for most problems ranges from 40 ~ 200, according to Xin-She [2021].
For each iteration of the Genetic Algorithm, the individuals that are in charge of generating the offspring of the current population are selected. These individuals are called the *parents* of the generation. Afterwards, the *children* of the population are generated, which is done by iteratively taking a pair of parents and applying the crossover operator to them with probability P_c; this parameter is the main operator of the algorithm and is usually set as nearly equal to 1, as stated by Swayamsiddha [2020]. After, each child is mutated with probability P_m; the mutation operator changes a gene of the child and is used for exploration of new solutions. The mutation probability P_m is usually set to be in the range 0.001 ~ 0.05, as claimed by Xin-She [2021].

Subsequently, based on its fitness, it is assessed if a child is part of the next generation or not. Only if a child has a better fitness than the individual with the worst fitness value of the population, the child replaces said individual. This is done for each child and the resulted population is the new generation. The steps above are repeated until a stopping criteria is met. Some common examples of termination conditions are: when the fitness value of the best solution has not improved for a fixed number of generations or when the number of generations is equal to a fixed number. The pseudo code of a GA can be seen in Algorithm 1.

Algorithm 1 Genetic Algorithm

```
population ← Initial Population
while termination condition is not met do
    parents ← Selection(population)
    offspring ← Crossover(parents,P_c)
    mutated_offspring ← Mutation(offspring,P_m)
    for i in mutated_offspring do
        z ← individual in population with worst fitness value
        if Fitness(i) is better than Fitness(z) then
            population ← population \ {z}
            population ← population ∪ {i}
        end if
    end for
end while
```

A disadvantage of GA is that several crucial parameters need to be set; the population size, the reproduction method, the probability of crossover and of mutation, and the stopping criteria, as maintained by Swayamsiddha [2020]. The decisions involved to define these parameters should be taken carefully, since any inappropriate choice of these variables can make a relevant difference in the quality of the solutions obtained.

4.1. Implementation of GA

In this section, it is thoroughly discussed how the GA was adapted to the problem of this thesis in order to solve it. The representation of an individual is defined, the main operators used in the algorithm are presented, as well as the parameters required to implement it.

4.1.1. Representation

The algorithm works with individuals that have two components; assuring nurses have feasible schedules and making sure the health care nurses provide is consistent with their skills. For the first component, the algorithm works with matrices S of size $|N| \times |T|$, where N is the set of nurses and T is the set of days considered in the time horizon. A gene of an individual is seen as a row in the matrix, where row $n \in N$ will show the schedule of nurse $n \in N$ throughout the time horizon T. In other words, it is a vector with $|T|$ entries showing for each day $t \in T$ whether a nurse works or not; if said nurse works on day $t \in T$, it states if the nurse is providing full-time health care and in which ward, or if the nurse is training on the job and in which ward, or if the nurse is attending advanced training and on which specialty.

In addition, to differentiate whether a nurse is working or training, and whether the training is on the job or if it is advanced training, it is proposed to assign a color to each entry. An entry will have color Red if the nurse is attending advanced training on the respective day, it will have color Green if the nurse is training on the job on the respective day, and it will have color Blue it the nurse is providing full-time health care on the respective day.
The following array gives an example of how a schedule of a nurse would look throughout the time horizon.

$$\overbrace{\begin{pmatrix} 0 & ONC & ONC & \cdots & 0 & GYN \end{pmatrix}}^{\text{Time Horizon } T}$$

This array states that with this schedule, a nurse would rest on the first day of the time horizon, he/she will attend advanced training for the Oncology specialty on the second day, on the third day, the nurse will provide full-time health care in the Oncology ward, and on the last day of the time horizon he/she will train on the job in the Gynecology ward.
The idea is to do this for each nurse, so that a schedule for all nurses would look like the following matrix:

$$S = \begin{array}{c} \overbrace{\begin{pmatrix} 0 & ONC & ONC & \cdots & 0 & GYN \\ GYN & GYN & 0 & \cdots & ICU & ICU \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & ONC & \cdots & ONC & ONC \end{pmatrix}}^{\text{Time Horizon } T} \end{array} \begin{array}{l} \rightarrow \text{Schedule for nurse 1} \\ \rightarrow \text{Schedule for nurse 2} \\ \qquad\vdots \\ \rightarrow \text{Schedule for nurse } |N| \end{array}$$

Furthermore, what makes a matrix feasible regarding the constraints for the nursing personnel depends on the schedules throughout the time horizon for each nurse. In other words, a matrix is feasible if the schedule for each nurse respects the number of days that said nurse must have worked at the end of the time horizon, if the minimum and maximum number of days each nurse should have worked each week is respected, if training on the job goes uninterrupted, if advanced training is taken only by authorized nurses, if nurses work on wards that are not their own only after finishing training on the job on the respective wards, and all the contractual constraints that nurses have.

The second component of a solution has to make sure the health care provided by nurses is consistent with their skills and with their schedules from the first component of the solution. For this second component, a matrix χ_n of dimension $|J| \times |T|$ is introduced for each nurse $n \in N$, where the entry $(\chi_n)_{jt}$ gives the hours of health care nurse $n \in N$ provides to patient type $j \in J$ on day $t \in T$. Hence, what makes a solution $\left(S, \{\chi_n | n \in N\}\right)$ feasible is if the matrix with schedules for all nurses S is feasible regarding the constraints for the nursing personnel, and it is consistent with the matrices χ_n for each $n \in N$.
In other words, if the matrix S states that nurse $n \in N$ is attending advanced training on day

$t \in T$, then $(\chi_n)_{jt}$ must be zero for all $j \in J$. If the matrix S states that nurse $n \in N$ is working in ward $w \in W$ on day $t \in T$, then $(\chi_n)_{jt}$ must be zero for all patient types $j \in J$ that are not in ward $w \in W$, it must be zero for all patient types $j \in J$ that require skills nurse $n \in N$ does not have, and the hours of health care nurse $n \in N$ provides on day $t \in T$ cannot be greater than eight hours, i.e. $\sum_{j \in J} (\chi_n)_{jt} \leq 8$. Finally, if the matrix S states that nurse $n \in N$ is not working on day $t \in T$, then $(\chi_n)_{jt}$ must be zero for all patient types $j \in J$.

If a solution $\left(S, \{\chi_n | n \in N\}\right)$ meets the described criteria, it is said it is a feasible solution.

4.1.2. Initial Population

The initial population used for the GA is generated randomly yet feasible: the matrices that describe the schedules for the nurses throughout the time horizon are generated first, and the distribution of health care among nurses to patients is deduced after, taking into account the already existing schedule.

For nurse $n \in N$, a schedule throughout the time horizon T is generated as follows:

1. The schedule for nurse $n \in N$ starts empty, i.e. not working on any day throughout the time horizon T.

$$\overbrace{\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}}^{\text{Time Horizon } T} \rightarrow \text{ Schedule for nurse } n$$

2. With probability $1/2$, it is decided whether nurse $n \in N$ takes advance training or not. If so, from the set of advanced trainings nurse $n \in N$ is eligible to take, one of them is randomly chosen and the days when said advance training take place are allocated into the schedule of nurse $n \in N$. If nurse $n \in N$ is not eligible to take any advance training, then the schedule continues to be empty.

$$\begin{pmatrix} 0 & AdvTrain & 0 & \cdots & AdvTrain & 0 \end{pmatrix}$$

3. For each week $i \in I$, an integer d_i that states how many days nurse $n \in N$ works in week $i \in I$ is randomly chosen. The number d_i has to respect the minimum and maximum number of days nurse $n \in N$ has to work weekly by contract, and it considers as well the scheduled days for the advanced training nurse $n \in N$ undertakes (if applicable).
   Then, d_i available days are randomly chosen in week $i \in I$, which will be the ones when nurse $n \in N$ will work. If by the end of the time horizon the total number of days nurse $n \in N$ works throughout the time horizon is greater than the number of days she is contractually obliged to work, a week where nurse $n \in N$ works more than the minimum weekly number of days is randomly selected. Afterwards, a day from this week where nurse $n \in N$ is scheduled to work is randomly selected and it is set to 0, i.e. this day is set to be a rest day for nurse $n \in N$. This is done until nurse $n \in N$ has reached the total number of days he/she is contractually obliged to work throughout the time horizon.
   The case where the total number of days nurse $n \in N$ works throughout the time horizon is smaller than the number of days she is contractually obliged to work is solved similarly. In the exemplification below, the gray squares represent the days where

nurse $n \in N$ is working.

$$\begin{pmatrix} \boxed{0} & 0 & \cdots & \boxed{0} & \boxed{0} & 0 & \cdots & 0 & \boxed{0} & 0 \end{pmatrix}$$

4. The set of days where nurse $n \in N$ works throughout the time horizon T is chronologically sorted and while said set is not empty, the following steps are done; the day $t \in T$ that chronologically follows that has not been scheduled yet is taken, and for this day $t \in T$ it is decided with probability p_1 if nurse $n \in N$ will provide full-time health care and with probability p_2 if nurse $n \in N$ will start training on the job in a ward where he/she has not already done so. Values p_1 and p_2 are chosen in a way so that $p_1 + p_2 = 1$, but $p_1 > p_2$; since the problem aims to minimize the size of flexible personnel, the probability for each nurse to become a flexible nurse is aimed to be small. Specifically for this thesis, $p_1 = 0.9$ and $p_2 = 0.1$.
   If it is chosen that nurse $n \in N$ is providing full-time health care on the fixed day $t \in T$, then it is taken into account the wards where said nurse can provide full-time health care. This is done by considering the ward $w_n \in W$ where nurse $n \in N$ belongs to, and the training nurse $n \in N$ has done on the previous days of the time horizon in other wards, and a ward $w \in W$ among these is randomly chosen. Then, the schedule of nurse $n \in N$ is updated so it shows that on day $t \in T$ nurse $n \in N$ provides full-time health care in ward $w \in W$.

$$\begin{pmatrix} \cdots & \boxed{\text{Ward w}} & 0 & \cdots & 0 & \boxed{0} & 0 \end{pmatrix}$$

   If on the other hand it is decided that nurse $n \in N$ starts training on the job on day $t \in T$, it is decided in which ward $w \in W$ the training can take place. This is done by considering the wards where nurse $n \in N$ has not done training on the job, and a ward $w \in W$ among these is randomly chosen. Let t_n^* be the number of days nurse $n \in N$ is working after day $t \in T$. Afterwards, the schedule of nurse $n \in N$ is updated so it shows that on day $t \in T$ nurse $n \in N$ starts training on the job in ward $w \in W$, and this training is also scheduled for the following $min(e_{nw}, t_n^*)$ days of the time horizon when nurse $n \in N$ is working; e_{nw} is the number of days it takes nurse $n \in N$ to be able to provide full-time health care in ward $w \in W$.

$$\begin{pmatrix} \cdots & \boxed{\text{Ward w}} & 0 & \boxed{\text{Ward w}} & \cdots & 0 & \boxed{\text{Ward w}} & \boxed{\text{Ward w}} & 0 & \boxed{0} \end{pmatrix}$$

This way, the schedules for all nurses are randomly generated yet feasible. Now, for a complete feasible solution, a distribution of the health care available that is consistent with the generated schedules is provided. To assure this, the following points are taken into account:

- If a nurse has a rest day or is attending advanced training, then he/she cannot provide any health care
- Patient types that require higher/more skills are assigned to be covered first
- A nurse can only provide health care to patients that are in the same ward the nurse is working in

- A nurse cannot provide health care to patients that require higher/different skills he/she has

Below are the steps on how this assignment is done.

1. For each day $t \in T$, the nurses that are working on day $t \in T$ and the amount of health care in hours they can provide are retrieved. Since the nurses attending advanced training cannot provide any health care, only the nurses that either are providing full-time health care or training on the job are considered.

2. Patient types are sorted so that the ones that require more/higher skills are assigned to be covered first, let $\hat{J}$ be the set of sorted patient types.

3. For each patient type $j \in \hat{J}$, the nurses that are working in the ward $w_j \in W$ where patient type $j \in \hat{J}$ belongs to are retrieved, and for each nurse $n \in N$ it is assessed if he/she has the required skills to provide health care to patient type $j \in \hat{J}$ or not (for this, it is taken into account if nurse $n \in N$ has acquired new skills with advanced training); if he/she can provide health care to said patient type, said nurse is set to provide the maximum amount of health care possible to this patient type. Every time, the available health care nurse $n \in N$ has and the health care patient type $j \in \hat{J}$ requires on day $t \in T$ is being updated.

4. By the end, the health care not met by the existing nursing personnel is assumed to be fulfilled by hiring external nurses.

This way, given a schedule for all nursing personnel, the available health care is distributed in a viable way and the demand not met by the existing nursing personnel is satisfied by external nurses. However, since the schedules are generated randomly, it is probable that the solutions generated in the initial population has plenty of idle nurses. Hence, the schedule for each individual of the initial population is modified in order to decrease idle nurses; these modifications aim to put the working days of each nurse where they are needed the most, while making sure the resulting schedules are still feasible and the scheduled trainings are respected.
Given a feasible solution, the following steps are done in order to possibly improve it:

1. For each week $i \in I$, and each nurse $n \in N$, the days on week $i \in I$ where the nurse does not provide any amount of health care (without considering days where the nurse attends advanced training), and the days of week $i \in I$ where the nurse does not work are retrieved.

2. For each patient type $j \in \hat{J}$, the days of week $i \in I$ where the needed health care of patient type $j \in \hat{J}$ is met by external nurses are retrieved.

3. For each day $s \in T$ when nurse $n \in N$ is idle, and for each day $t \in T$ where nurse $n \in N$ is not working and there is demand not met by the existing nursing personnel, it is first checked if nurse $n \in N$ has the required skills to provide health care to patient type $j \in \hat{J}$. If so, then the schedule nurse $n \in N$ has on days $s \in T$ and $t \in T$ may be rescheduled.

4. It is important to notice that this change depends on what nurse $n \in N$ is doing on day $s \in T$; if the nurse is training on the job on the same ward where patient type $j \in \hat{J}$ is, then the training on the job nurse $n \in N$ takes in said ward is rescheduled so that the nurse works on day $t \in T$ and covers some amount of health care patient type $j \in \hat{J}$ needs. If on the contrary, on day $s \in T$ nurse $n \in N$ is training on the job in a different ward than the one patient type $j \in \hat{J}$ is, then the schedule for said nurse on day $s \in T$ does not change since training on the job has to be uninterrupted.
   If nurse $n \in N$ is providing full-time health care on day $s \in T$, nurse $n \in N$ will work on day $t \in T$ instead of day $s \in T$ in the ward $w_j \in W$ where patient type $j \in \hat{J}$ is, if said nurse either belongs to ward w_j or has finished training on the job in ward $w_j \in W$ by day $t \in T$, and if training on the job does not get interrupted by providing full-time health care on day $t \in T$.

This procedure takes the schedules randomly generated and aims to accommodate the working days of each nurse where they are needed the most, as long as the resulting schedule still respects the contractual constraints nurses have and the trainings already scheduled for nurses.

4.1.3. Fitness Evaluation

To compute the fitness for each solution in the population, the objective function presented in Chapter 3 is used. This function considers the amount of time spent in training on the job, the number of flexible nurses, the hours of health care that the hospital must provide by external nurses, and the training on the skills for which priority training is applied. Given a feasible solution, all of the previous factors are computed and each solution gets a fitness value.

4.1.4. Reproduction/Selection Operator

This operator selects the solutions of the population that will be the parents of the current generation's offspring. Two approaches are implemented for this operator: a rank based selection and a roulette wheel selection. To exemplify these, integer parameters a and b are given.
For the rank based selection, the individuals of the population are ordered by their rank; best, second best, third best, etc. And they are weighted based on their rank, so the best individual is chosen with probability $1/2$, the second best with probability $1/3$, and the i^{th} best solution is chosen with probability $1/(i+1)$. With these probabilities, b solutions are chosen.
For the roulette wheel selection, a sample of individuals of size a is randomly selected from the population and the solution with the best fitness score is selected to be a parent. This procedure is done until the number of parents is b.
These two selection operators are combined by implementing the rank based operator with probability P_{RB}, and applying the roulette wheel operator with probability $1 - P_{RB}$.

4.1.5. Crossover Operator

Once the parents have been selected, the uniform crossover operator is used with probability P_c. This operator generates an offspring by copying genes from parents.
Parents are matrices that describe schedules for all nurses, and the genes that are being

inherited are seen as nurses' schedules. The parent from which each gene is copied is determined by a *crossover mask*. A crossover mask is a $\{0,1\}^{|N|}$ vector, and if in the n^{th} entry there is a 1, the respective gene (in this case the schedule for the nurse $n \in N$), is copied from the first parent, and if there is a 0, said gene is copied from the second parent. An example of the crossover operator implemented is shown in Figure 4.1.

Figure 4.1: Graphical example of how the crossover operator works.

4.1.6. Mutation Operator

Once a child is generated, one of its genes can be changed with probability P_m. This is done with the purpose of keeping diversity in the population. The mutation operator proposed is a bottleneck heuristic that aims to create a whole new schedule for the gene (nurse) chosen. If a solution is decided to be mutated, the operator chooses randomly either the nurse that has the most idle time, or any other nurse; and the schedule of the chosen nurse is the one that will be changed; say it is nurse n. The schedules of the rest of the nurses are fixed and the ILP formulation that considers the health care that has not been met and only nurse n is solved using the solver Gurobi and the new schedule for nurse n is retrieved from the solution.

Since the mutation operator uses Gurobi, it is possible that for some instances applying this operator is no longer feasible since Gurobi may take an impractical amount of time to solve the restricted instance. For these instances, a time limit within Gurobi can be used, so if Gurobi exceeds said time limit while solving an instance, the solve is interrupted and the mutation operator does not take place. However, a modified mutation operator is proposed for these instances; for a fixed nurse $n \in N$, the operator described above is applied for the first t days in the time horizon, and for the remaining $|T| - t$ days, the schedule for nurse $n \in N$ is randomly built in the same way the initial solutions were generated, taking into account the optimal schedule for the first t days provided by Gurobi. The generated schedule for the last $|T| - t$ days is then improved following the same algorithm used to improve the initial solutions.

This modified mutation operator was defined like this since the ILP formulation of the model does not consider the unfinished training on the job that has started before the time horizon, hence, the small instance aimed to solve with Gurobi is taken considering the beginning of the time horizon.

It is important to notice that given the structure of the solutions, and the definition of the operators used, the proposed GA will work with feasible solutions throughout the implementation of the algorithm.

4.1.7. Stopping Criteria

To determine when to stop the algorithm, two termination conditions are considered so that the algorithm stops whenever one the two stopping criteria is met first.
Let l and m be integer values. The Genetic Algorithm will stop if the population has not improved for the previous l generations, or if the number of generations that have been generated is equal to m. Combining these two stopping criteria helps to stop the algorithm once the population has converged, saving execution time of the algorithm.

5

Instance based on Erasmus MC

This chapter presents the data that is used to test the model presented in Chapter 3 and the solution method described in Chapter 4. The instance presented below is based on real data provided by the hospital Erasmus MC.
Overall, 141 nurses, 4 wards, 10 patient types, and 3 different types of advanced training are considered. The following sections provide a description of every part of information that was required from the hospital.

5.1. Contractual Obligations

On average nurses work 32 hours per week (4 shifts per week). Since the hospital works with annual hours, it may be possible for a nurse to work 24 hours in one week and up to 40 hours in another week. This means that on average a nurse can work between 3 to 5 shifts per week, taking care that the nurse meets the annual average of 4 shifts per week.
Furthermore, on average, the cost of hiring an external nurse per hour is 55 euros.

5.2. Hospital's Demand

The hospital's demand that must be fulfilled is seen as the amount of health care (in hours) that each patient type requires per day. A patient type is defined by two components: the ward the patient belongs to, and the specialty said patient enters to.
Erasmus MC provided the data of all patients that the hospital admitted through the year 2020; from this, only data from the considered wards was extracted. For each day, the amount of time each patient spent in the hospital was retrieved and it was added to the respective patient type's total amount of health care (in hours) required.

5.3. Advanced and Priority Training

Three types of advanced training are considered:

- **Oncology specialty**. This training lasts 13 months, nurses undertaking this training go to school 3 days each month, and for these days nurses cannot provide any health care.

- **Traumatology specialty**. This training lasts 3 months, nurses undertaking this training go to school 3 days each month, and for these days nurses cannot provide any health care.

- **BS (Acute health care) specialty**. This training lasts 3 months, nurses undertaking this training go to school 2 days each month, and for these days nurses cannot provide any health care.

It is important to notice that these trainings may not be mandatory for nurses to be able to provide health care to patients in this specialization. Nurses in the hospital's Gynecology - Urology ward that do not have the *BS* specialty, for example, are able to provide health care to emergency patients. However, nurses with this advanced training are able to provide a more efficiently health care to these type of patients. Furthermore, due to the COVID-19 pandemic, the BS specialty training is encouraged by the hospital for nurses to take, so it will be the specialty considered as priority training.

5.4. Time Horizon

The time horizon considered is of 364 days (52 weeks), starting from Monday, January 6, 2020, until Sunday, January 2, 2021.

5.5. Wards

This section provides the required data for each ward, such as: the total number of nurses the ward has, the number of nurses that have advanced training, the type of advanced training nurses can take in each ward, the patient types each ward treats, and the amount of time a nurse from another ward has to train on the job until said nurse can provide full-time health care in the ward.

5.5.1. Sophia URO - GYN

This ward is structured as follows:

- The ward consists of 30 nurses, from which 6 of them have advanced training in the Oncology specialty.
- Nurses from this ward can take the Oncology specialty training and the BS specialty training.
- There are two patient types: patients admitted in the Gynecology specialization and the ones admitted in the Urology specialization.
- On average, nurses from another ward take one month of training on the job until they are able to provide full-time health care in this ward.

5.5.2. Dijkzigt Beweging

This ward is structured as follows:

- The ward consists of 42 nurses, from which 15 of them have advanced training in the Traumatology specialty.
- Nurses from this ward can take the Traumatology specialty training.

- There are three patient types: patients admitted in the Traumatology specialization, the ones admitted in the Plastic Surgery specialization, and the ones admitted in the Orthopedics specialization.
- On average, nurses from another ward take one month of training on the job until they are able to provide full-time health care in this ward.

5.5.3. Daniel Oncologie

This ward is structured as follows:

- The ward consists of 39 nurses, from which 8 of them have advanced training in the Oncology specialty.
- Nurses from this ward can take the Oncology specialty training.
- There is one patient type: patients admitted in the Oncology specialization.
- On average, nurses from another ward take five months of training on the job until they are able to provide full-time health care in this ward.

5.5.4. Daniel Hoofd Hals

This ward is structured as follows:

- The ward consists of 30 nurses.
- The ward does not provide advanced training.
- There are four patient types: patients admitted in the Oral, Maxillofacial and Facial Surgery specialization, the ones admitted in the Ear-Nose-Throat Surgery specialization, the ones admitted in the Ophthalmology specialization, and the ones admitted in the Radiotherapy specialization.
- On average, nurses from another ward take five months of training on the job until they are able to provide full-time health care in this ward.

6

Results

This chapter displays the results of solving different instances based on the data discussed in Chapter 5. These instances are solved by two approaches; with the Integer Linear Programming (ILP) formulation shown in Chapter 3 using Gurobi, and the Genetic Algorithm introduced in Chapter 4; the performance of each approach is assessed and a comparison between both approaches is done. Experiments were run on a computer with a 2.20GHz Intel Core i7-8750H processor running with 16 GB of RAM.

Table 6.1 displays the instances that were contemplated. The differences between the instances shown are the number of wards considered and the time horizon. The number of nurses and the number of patient types is given by the wards contemplated in each instance. The first three instances have the same time horizon of 28 days (~ 1 month), instance 1 considers wards Sophia URO - GYN and Daniel Oncologie, instance 2 adds ward Dijkzigt Beweging to instance 1, and finally instance 3 considers all four wards. Instance 4 considers all four wards and a time horizon of 84 days (~ 3 months), and instances 5 and 6 consider all four wards, but instance 5 takes a time horizon of 182 days (~ 6 months), and instance 6 takes a time horizon of 364 days (1 year).

Instance ID	# Wards	# Nurses	# Patient types	Days in time horizon
1	2	69	3	28
2	3	111	6	28
3	4	141	10	28
4	4	141	10	84
5	4	141	10	182
6	4	141	10	364

Table 6.1: Instances considered for testing Gurobi and the GA.

Table 6.2 shows the result of each instance using Gurobi. It can be seen that Gurobi could solve in a reasonable time instance 1 to optimality. For the remaining instances, either Gurobi was taking impractical running time so it had to be manually interrupted, or the Branch & Bound tree implemented by Gurobi exhausted all available main memory.

For instances $2-4$, Gurobi provides upper and lower bounds for the value of an optimal solution in the objective function, for instance 5, Gurobi could only provide an upper bound (best integer/feasible solution found by Gurobi), and for instance 6, Gurobi could not provide any bound.

Instance ID	Optimal Solution Found	Running Time	Comments
1	YES	8min 31secs	-
2	NO	4h 39min 20sec	Manually Interrupted
3	NO	10h 0min 0sec	Manually Interrupted
4	NO	13h 0min 0sec	Manually Interrupted
5	NO	23min 13sec	Gurobi Error: Out of memory
6	NO	20min 30sec	Gurobi Error: Out of memory

Table 6.2: Results using Gurobi.

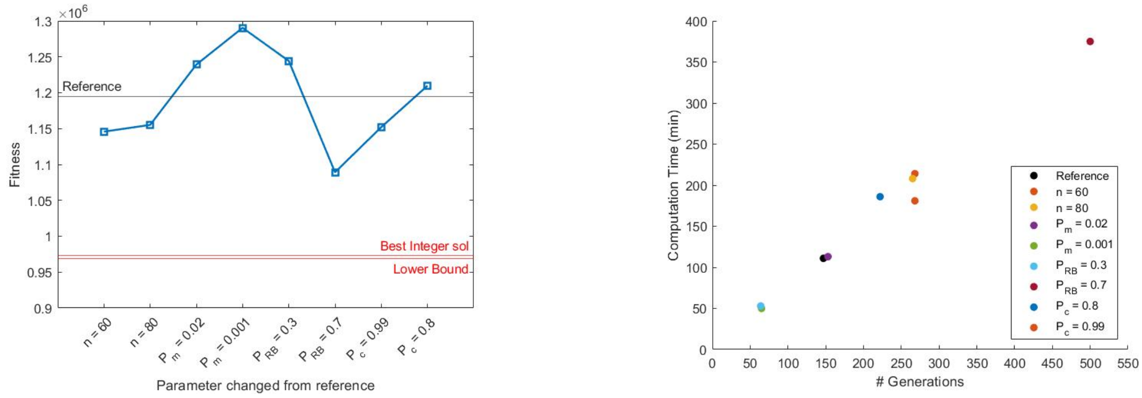
Tables A.1 - A.6 show the results of implementing the Genetic Algorithm to instances $1-5$. For each of these, the Genetic Algorithm was implemented several times, each with different parameters; the parameters used as a reference are the following:

- Population size (n) of 40
- Probability of mutation (P_m) of 0.05
- Probability of applying the rank based selection operator (P_{RB}) of 0.5
- Probability of crossover (P_c) of 0.9

Every run of the Genetic Algorithm to an instance would take the parameters used as reference and slightly change one of them; the parameters and the changes done to them in each run were based on what the literature suggested. For example, Xin-She [2021] states that the probability of applying the crossover operator should be close to 1, and the probability of applying the mutation operator should not be bigger than 0.05.

Figure 6.1 visually represents the results from implementing the GA to instance 3. Figure 6.1a displays the fitness value of the best solutions from the last generations of each experiment. The fitness value of the solution obtained from using the GA with the reference parameters is shown with a horizontal black line. The upper and lower bounds provided by Gurobi are displayed as horizontal red lines. It can be seen that the experiment with the best fitness value was the one obtained by increasing P_{RB} to 0.7 from the reference parameters.

Figure 6.1b shows, for each experiment, the number of iterations performed in each one and the computation time in minutes it took to do so. For the experiment where parameter P_{RB} was increased to 0.7, it took the longest to implement the GA than the rest of the other runs.
Figures A.1, A.2, and A.3 display similar visual representations of the obtained results for instance 1, instance 2, and instance 5, respectively.



(a) Fitness of final solution

(b) Number of generations versus computing time

Figure 6.1: Visual results of applying the GA to instance 3.

For instances $1-3$ Gurobi outperformed the proposed GA since it provided a better feasible solution. For these instances, the Genetic Algorithm stopped when the population did not improve during 10 consecutive generations, or when the number of generations generated was equal to 500. It can be seen that only one implementation of the GA to instances $1-3$ reached the 500 generations; the one that yield the best solution obtained for instance 3, for the rest the respective population converged before reaching the 500 generations.

Furthermore, while testing the GA for instance 4, it was noticed that the implementation of Gurobi (without any time limit) within the mutation operator started to take too much time, hence for this instance the GA was implemented with both the mutation operator having a time limit for Gurobi of 120 seconds, and with the modified Mutation operator proposed in Section 4.1.6 with parameter t set to 56 ($\sim$ 2 months). This was done with the purpose to compare the results of the GA with a time limit on Gurobi within the mutation operator and by implementing the modified mutation operator.

For instance 4 Gurobi outperformed the proposed GA since it provided a better feasible solution. For this instance, the Genetic Algorithm stopped when the population did not improve during 10 consecutive generations, or when the number of generations generated was equal to 300. Figure 6.2 shows a comparison of the results obtained by the GA with a time limit on Gurobi within the mutation operator and by implementing the modified mutation operator. It shows that there is not one approach that performed better for all experiments. However, Tables A.4 and A.5 show that the GA when applying a time limit on Gurobi, most of the times the population converges faster and there are less generations done; which makes sense since less mutations are done to the offspring generated due to the time limit within Gurobi, and the population had less chance of maintaining variety (which is the purpose of the mutation operator within the GA).
Consequently, for instances 5 and 6, the GA with the modified mutation operator is implemented.

Moreover, Gurobi started to have problems when trying to solve instance 5; Gurobi interrupted itself due to memory issues and it only provided an upper bound for the optimal solution of the instance. However, said bound was quite easy to improve since the fitness of the best solution of the initial population for each experiment was already better than

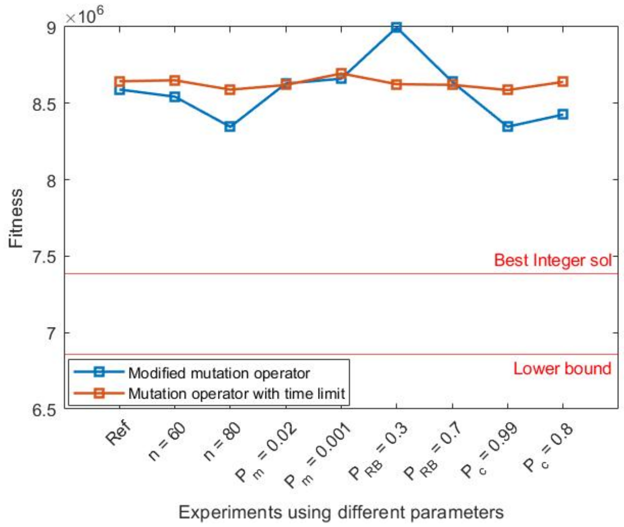


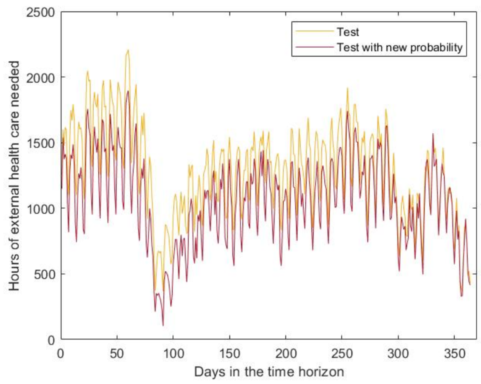
Figure 6.2: Visual comparison between the implementation of the GA with the modified mutation operator and the GA with the mutation operator with time limit, both applied to instance 4.

the bound provided by Gurobi.

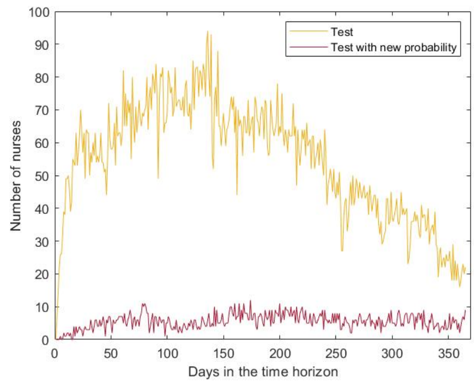
Since instance 5 considers twice the amount of days in the time horizon as instance 4, it was decided to implement the Genetic Algorithm with the modified mutation operator with parameter t set to 56 ($\sim$ 2 months) for instances 5 and 6.
For instance 5, the GA stopped when the population did not improve during 10 consecutive generations, or when the number of generations generated was equal to 120. It can be seen in Table A.6 that five out of the nine experiments done for this instance performed 120 generations, hence for these experiments the GA could have kept improving the population if more generations were allowed; however, this would have taken more computation time.

For the results that were obtained by implementing the GA for instances $1-5$, it is evident that parameters like the population size, the probability of crossover and of mutation, play a crucial part in the solutions obtained, and the choice of these parameters should be chosen carefully.

To solve instance 6 by implementing the GA, some experiments were implemented considering a small population size and a small number of generations for the stopping criteria. It was noticed that with a population size of 10 and performing 15 generations, the following test solution was obtained: 53 nurses take the priority training BS (Acute health care advanced training), and all 141 nurses take at least one training on the job in another ward, hence all the nursing personnel is flexible at the end of the time horizon.
Even though this is a feasible scenario, it is not realistic and it is quite expensive; since the implementation of the GA is expected to take a considerable computation time to run, it is proposed to start with a better solution; as stated in Section 4.1.2, when it is decided if a nurse provides full-time health care or if he/she starts training on the job, the nurse will start training on the job with probability of 10%. Once it is decided said nurse will start training on the job, a ward where he/she can do this is randomly selected and the nurse starts training on the job in the chosen ward.



(a) Comparison on external health care needed throughout the time horizon



(b) Comparison on number of nurses training on the job throughout the time horizon

Figure 6.3: Comparison of the test solution and the test solution with new probability of training in another ward.

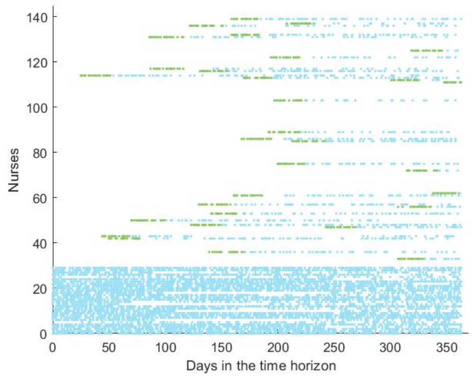
What is proposed is to assign a probability to start training on the job in each ward: once it is decided nurse $n \in N$ can start training on the job, a ward $w \in W$ where he/she can do this is randomly selected and said nurse will start training on the job in ward $w \in W$ with probability $\frac{1}{e_{nw}}$, where e_{nw} is the number of days it takes nurse $n \in N$ to finish training on the job in ward $w \in W$.

With this new probability added, another test solution was generated with the same parameters as the one before and this was the result: 46 nurses take the priority training BS (Acute health care advanced training), and 59 nurses take at least one training on the job in another ward, hence the hospital will have 59 flexible nurses at the end of the time horizon. Figure 6.3 shows further comparison between both solutions. Figure 6.3a shows the external health care in hours needed to fulfil the hospital's demand throughout the time horizon in both solutions; it can be seen that the solution obtained with the changed probability requires less hours of external workforce.
In addition, Figure 6.3b shows the number of nurses that are taking training on the job on each day in the time horizon, and as expected, the number of nurses training on the job throughout the time horizon in the solution with the changed probability is smaller.

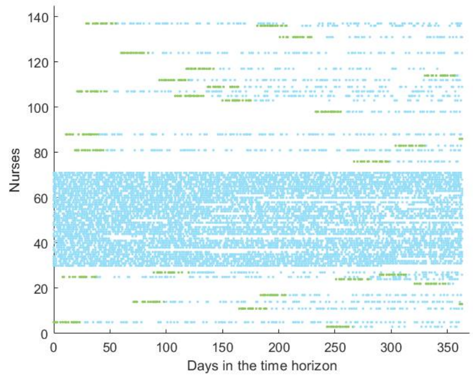
Instance 6 is solved by using the genetic algorithm with this changed probability on taking training on the job, and with the modified mutation operator. The following parameters are chosen:

- Population size (n) of 30
- Probability of mutation (P_m) of 0.05
- Probability of applying the rank based selection operator (P_{RB}) of 0.5
- Probability of crossover (P_c) of 0.99

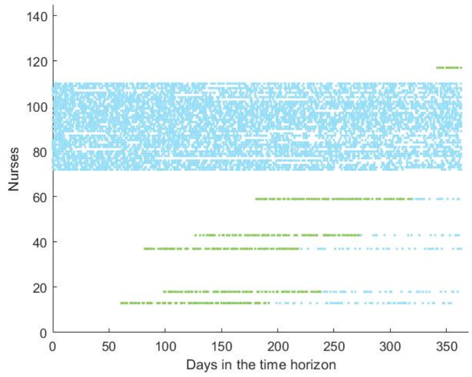
The GA stopped when the population did not improve during 10 consecutive generations, or when the number of generations generated was equal to 20. The computation



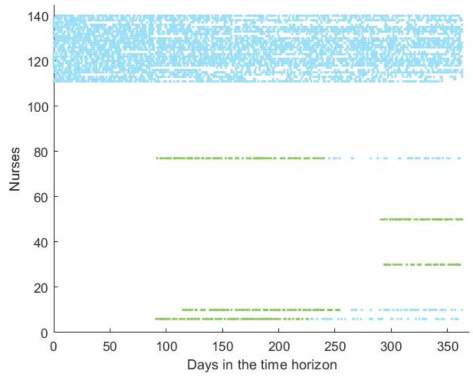
(a) Days when nurses took training on the job (green) and days when nurses provided full-time health care (blue) in ward Sophia URO - GYN



(b) Days when nurses took training on the job (green) and days when nurses provided full-time health care (blue) in ward Dijkzigt Beweging



(c) Days when nurses took training on the job (green) and days when nurses provided full-time health care (blue) in ward Daniel Oncologie



(d) Days when nurses took training on the job (green) and days when nurses provided full-time health care (blue) in ward Daniel Hoofd Hals

Figure 6.4: Schedules of nurses for each ward throughout the time horizon.

time it took to run was 8 hours, 25 minutes and 52 seconds, it performed 20 generations in total, and the population had not converged yet. The fitness value of the best solution of the initial population is 21.225.321, and the fitness value of the best solution in the last generation is 20.982.083; the genetic algorithm improved the best initial solution by 1%. With the obtained solution, 38 nurses take priority training, and 51 nurses take training on the job in another ward at least once, hence there are 51 flexible nurses at the end of the time horizon.

Figure 6.4 shows a graphical representation of the solution: for each ward, it shows the days of the time horizon when nurses took training on the job or provided full-time health care in said ward. It can be seen that the nurses belonging to the ward represented do not take training on the job, and nurses from a different ward provide full-time health care only after finishing training on the job.

7
Discussion & Recommendations

This thesis aimed to create a mathematical model that could help Erasmus MC in the workforce planning process for a fixed time horizon in order to generate a flexible workforce as small as possible, and encouraging priority training while minimizing the costs to fulfill the hospital's demand; these costs include hiring external nurses and the time nurses spent on training on the job.
This chapter discusses the decisions that were made during this project, the limitations this thesis has and further recommendations to follow up on the research and implementation done.

7.1. Mathematical Formulation

The first part of the thesis was to model the problem as an Integer Linear Programming (ILP) problem. For this, the model for the management skills problem provided by DePuy et al. [2006] was taken as the starting point since it was the work found that was the closest to this thesis' objectives. However, several things needed to be changed and/or added for the model presented by DePuy et al. [2006] to be exactly what this thesis aimed to accomplish.

First of all, the model shown by DePuy et al. [2006] does not have a breakdown on how or when training should be implemented among the workforce. A time horizon was needed, and at first a weekly time horizon was considered. However, with a weekly time horizon there would still be further decisions to be made, like which days of the week should the nurse work and/or train. For this reason, it was decided to have a daily time horizon; by considering a daily time horizon further constraints had to be taken into account. These constraints were linked to the contractual obligations of nurses with the hospital, such as the number of days each nurse can work in a week and how many days a nurse must have worked at the end of the time horizon.
With this arrangement of the time horizon, there are still further decisions to be made; for example, which shift should a nurse work on a day he/she is scheduled to work. Nevertheless, by considering shifts into the problem, further constraints must be added. However, the problem would become a rostering problem, which was not the problem aimed to solve

in this thesis.

Secondly, the model for this thesis had to consider the two types of training there are in Erasmus MC: advanced training and training on the job. Advanced training has fixed dates and the only decision to be made is whether a nurse takes this training or not (this training is the type of training considered by DePuy et al. [2006]), and training on the job is taken within the hospital and it does not have fixed dates; hence there are several decisions to be made for this type of training: if a nurse takes it or not, when does the nurse start said training, which ward does the training take place, and which days should he/she take the training. The latter was not considered in the literature found for this project, so decision variables and constraints were developed to model this type of training; these assure training on the job goes uninterrupted, they make sure a nurse must finish training on the job in another ward before being able to provide full-time health care in said ward, and they consider the fraction of health care a nurse can provide while training on the job.

Finally, variables had to be introduced in the model to keep track of the flexible nurses and the external personnel needed in order to fulfill the hospital's demand.
The resulting model presented in Chapter 3 has the following limitations:

- The parameter e_{nw} that states the time it takes for a nurse $n \in N$ to finish training on the job in ward $w \in W$, does not change during the time horizon; because of this, the model overestimates this measure of time. Once a nurse has completed training on the job in a ward, the time needed for the nurse to train on the job differs among wards. Hence, an option to fix this overestimation is to introduce constraints to the model that update the parameter e_{nw} depending on the training nurse $n \in N$ takes throughout the time horizon.

- The model does not consider the unfinished training on the job that has started before the time horizon. Because of this, there is an overestimation of the amount of health care available from the hospital's nursing personnel. This can be fixed by ensuring these nurses finish their training on the job before providing full-time health care, and to keep track how many days of training nurses have left, so the fraction of health care said nurses can provide is computed accordingly.

First, the ILP formulation of the model was tested using the mathematical optimization solver Gurobi with a small instance to assure the model was working as expected. Afterwards, the model was tested with bigger instances considering the real data provided by the hospital, and Gurobi started to struggle to solve some instances. Either the running time Gurobi was taking to solve the instance started to become impractical, or Gurobi itself would interrupt the solve with error code 10001; which means that the Branch & Bound tree has exhausted all available main memory.

Since the aim of this thesis was to solve the problem considering all the data provided by the hospital described in Chapter 5, a different solution method was needed in order to accomplish this.

7.2. Solution Method

The second part of this thesis was to develop a different solution method to solve the problem presented, which was needed since Gurobi was not able to solve it. Among all the different solution methods that were found in the literature, a Genetic Algorithm approach was chosen since it has shown to perform quite well in solving assignment and workforce planning problems according to De Bruecker et al. [2015]. The values of the parameters considered in the Genetic Algorithm were chosen by following the work of Xin-She [2021], Swayamsiddha [2020], and De Jong [1988].

The hardest challenge to implement a Genetic Algorithm approach was to figure out how a solution should be structured in order for it to be a feasible solution and to be able to apply the different operators the GA considers.
Furthermore, since the initial population is generated randomly, it was observed that the fitness of the resulting solutions in this initial population was not as good as expected. For this reason, it was decided to improve the randomly generated solutions by distributing the nurses where they were needed the most, as long as the changes resulted in a feasible solution and the trainings taken throughout the time horizon were respected.

Moreover, the mutation operator considered is a bottleneck heuristic that implements Gurobi to a smaller instance than the one aimed to solve. However, when solving instances that considered all four wards and a time horizon bigger than or equal to 84 days (~ 3 months), the mutation operator started to take impractical running time to finish. To solve this, the mutation operator was modified for these bigger instances (as explained in Section 4.1.6).

This modified mutation operator was defined this way since the ILP formulation of the model does not consider the unfinished training on the job that has started before the time horizon. Hence, the small instance aimed to solve with Gurobi is taken considering the beginning of the time horizon.
By having an ILP that does take into account unfinished training on the job, the modified mutation operator could then choose critical weeks/months from the time horizon and use Gurobi to provide the optimal schedule for the respective nurse in the chosen weeks/months.

Furthermore, while testing the GA to the bigger instances (specifically, instances 4 and 5), the running time needed was more than expected; so it was decided that the number of generations allowed decreased for each one of the instances. Since the stopping criteria on the maximum number of generations allowed decreased from instance 4 onwards, some implementations of the GA for these instances were stopped before the population could converge. This means that by allowing to run the GA for more generations, the solution obtained from the GA could have been improved.
In addition, comparisons between the performance of Gurobi and the GA were not always fair since the running time of both approaches were not the same. For example, for instance 3 Gurobi ran for 10 hours, while the GA ran for at most 6 hours. This can overestimate the error of the solutions of the GA when compared with the solution obtained by Gurobi; by letting the GA run for more time, the solution by the GA would have kept improving.

Moreover, several runs of the GA were implemented to instance 6 considering small parameters for the population size and for the maximum number of generations allowed, it was noticed that some solutions were implementing training on the job to all nurses in all the wards (except their own ward). So at the end all the nurses were flexible nurses. This is probably due to the way the initial population is generated, and the way the initial population is aimed to be improved.

Since improvement in the initial population respects the training done throughout the time horizon, training on the job that a nurse was assigned to do when generating random schedules was not changed when solutions were aimed to be improved. However, since the mutation operator used Gurobi, the trainings assigned to nurses were changed if they were not convenient; but Gurobi was only implemented for the entire time horizon in small instances. For the big instances, specifically for the instance aimed to be solved, Gurobi would only modify the first two months of nurses' schedules, and for the rest of the time horizon the schedule was generated randomly and improved in the same way as it is done when generating the initial population. This could be the reason most nurses were assigned to take training on the job in at least one other ward.

To prevent this, the probability a nurse could start training on the job in a ward when generating the initial population was changed. With this new probability, the number of flexible nurses at the end of the time horizon decreased when some runs were done, and the hours of external nurses needed decreased as well. This modification was done with the purpose of improving the initial population generated.

However, only the instance considering all data provided by the hospital was solved using this new approach, hence further research and testing should be done to the rest of the instances using the GA with this change.

The quality of the solutions obtained with the Genetic Algorithm could be improved by testing different parameters than the ones proposed; it may be that different parameters would yield better solutions. Moreover, solutions can also be improved by implementing further heuristics to the solution given by the GA. Also, a different way to generate the initial population can be implemented, so nurses are scheduled to work when they are most needed to start the GA with a stronger population. Nevertheless, the Genetic Algorithm implemented in this thesis outperformed Gurobi for the two biggest instances considered in this thesis.

As stated in Section 7.1, the mathematical formulation of the problem did not consider breaking the time horizon into shifts since a considerable amount of constraints should have been added and it would have become a rostering problem. However, breaking the time horizon into shifts with the presented Genetic Algorithm approach would have only implied one more feasibility check when generating feasible schedules for nurses. Hence, solving a rostering problem with the GA proposed is left as future research.

8
Conclusion

This thesis aimed to aid Erasmus MC in the workforce planning process of deciding how to implement training among the nursing personnel in order to build a more flexible workforce, so the hospital can be prepared for fluctuating and unexpected demand, while minimizing the costs of doing so.

To accomplish this, a mathematical model was developed taking into account the structure of the nursing personnel within the hospital, and the structure of how training among nurses is implemented. Since the planning was aimed to be done on a daily basis, the contractual obligations between nurses and the hospital had to be respected as well; like the number of hours a nurse can work in one week, for example.

The resulted model was tested using real data provided by Erasmus MC, and with the mathematical optimization solver Gurobi. However, for some instances, the implementation of the solver was not feasible anymore, since the running time Gurobi was taking to solve the instance started to become impractical, or Gurobi itself would interrupt the solve due to main memory issues. Consequently, a Genetic Algorithm (GA) approach was proposed; this heuristic accomplished to outperform Gurobi when applied to the biggest 2 instances considered in this project. Furthermore, the GA was tested for different instances using distinct parameters and comparing different operators.

The mathematical model and the Genetic Algorithm proposed in this thesis aim to provide, if not an optimal, a feasible schedule of the nursing personnel in a fixed time horizon in order to implement training in an efficient way to build a more flexible workforce; which would help Erasmus MC to be more prepared towards fluctuating and unexpected demand.

A

Appendix

Results of implementing the Genetic Algorithm for each of the instances shown in Table 6.1 are displayed below.

Genetic Algorithm: **Instance 1**									
	Population size (n)	P_m	P_{RB}	P_c	Running Time	# Generations	Best Initial Solution	Best Final Solution	Optimal Value
Reference	**40**	**0,05**	**0,5**	**0,9**	**39min 27sec**	**161**	**1.261.974**	**1.115.672**	1.062.530
Perturbing n	60	0,05	0,5	0,9	23min 11sec	78	1.257.423	1.141.925	
	80	0,05	0,5	0,9	39min 8sec	138	1.263.497	1.135.469	
Perturbing P_m	40	0,02	0,5	0,9	16min 26sec	58	1.247.043	1.155.443	
	40	0,001	0,5	0,9	20min 55sec	81	1.244.549	1.157.595	
Perturbing P_{RB}	40	0,05	0,3	0,9	20min 51sec	76	1.251.609	1.128.680	
	40	0,05	0,7	0,9	18min 21sec	66	1.254.605	1.148.710	
Perturbing P_c	40	0,05	0,5	0,99	38min 37sec	181	1.209.403	1.123.712	
	40	0,05	0,5	0,8	27min 40sec	169	1.213.038	1.168.879	

Table A.1: Results using the Genetic Algorithm on Instance 1, testing different parameters.

Genetic Algorithm: **Instance 2**									
	Population size (n)	P_m	P_{RB}	P_c	Running Time	# Generations	Best Initial Solution	Best Final Solution	Best Integer Solution by Gurobi / Lower Bound
Reference	**40**	**0,05**	**0,5**	**0,9**	**1h 35min 7sec**	**180**	**2.113.701**	**1.948.234**	1.774.202 / 1.769.654
Perturbing n	60	0,05	0,5	0,9	1h 38min 43sec	169	2.103.960	1.926.406	
	80	0,05	0,5	0,9	3h 36min 5sec	397	2.109.950	1.890.083	
Perturbing P_m	40	0,02	0,5	0,9	1h 22min 2sec	165	2.109.520	1.961.986	
	40	0,001	0,5	0,9	36min 14sec	65	2.094.845	1.999.228	
Perturbing P_{RB}	40	0,05	0,3	0,9	1h 35min 11sec	179	2.112.404	1.932.660	
	40	0,05	0,7	0,9	2h 45min 4sec	315	2.115.067	1.945.462	
Perturbing P_c	40	0,05	0,5	0,99	2h 19min 18sec	239	2.094.601	1.890.774	
	40	0,05	0,5	0,8	38min 6sec	72	2.104.014	1.984.843	

Table A.2: Results using the Genetic Algorithm on Instance 2, testing different parameters.

Genetic Algorithm: **Instance 3**									
	Population size (n)	P_m	P_{RB}	P_c	Running Time	# Generations	Best Initial Solution	Best Final Solution	Best Integer Solution by Gurobi / Lower Bound
Reference	**40**	**0,05**	**0,5**	**0,9**	**1h 51min 52sec**	**147**	**1.425.769**	**1.194.916**	973.270 / 969.078
Perturbing n	60	0,05	0,5	0,9	3h 34min 52sec	268	1.424.498	1.145.762	
	80	0,05	0,5	0,9	3h 28min 58sec	265	1.403.084	1.155.142	
Perturbing P_m	40	0,02	0,5	0,9	1h 53min 25sec	153	1.396.158	1.239.213	
	40	0,001	0,5	0,9	50min 46sec	65	1.396.514	1.290.359	
Perturbing P_{RB}	40	0,05	0,3	0,9	53min 23sec	64	1.389.581	1.244.156	
	40	0,05	0,7	0,9	6h 15min 11sec	500	1.403.202	1.089.151	
Perturbing P_c	40	0,05	0,5	0,99	3h 6min 46sec	222	1.403.548	1.152.023	
	40	0,05	0,5	0,8	3h 1min 54sec	268	1.420.128	1.209.700	

Table A.3: Results using the Genetic Algorithm on Instance 3, testing different parameters.

Genetic Algorithm with modified mutation operator: **Instance 4**									
	Population size (n)	P_m	P_{RB}	P_c	Running Time	# Generations	Best Initial Solution	Best Final Solution	Best Integer Solution by Gurobi / Lower Bound
Reference	**40**	**0,05**	**0,5**	**0,9**	**7h 50min 48sec**	**147**	**8.939.632**	**8.589.981**	7.387.929 / 6.860.623
Perturbing n	60	0,05	0,5	0,9	14h 55min 48sec	170	8.946.345	8.543.092	
	80	0,05	0,5	0,9	12h 27min 0sec	300	8.971.235	8.345.750	
Perturbing P_m	40	0,02	0,5	0,9	8h 11min 56sec	145	8.986.958	8.629.308	
	40	0,001	0,5	0,9	2h 37min 49sec	69	8.932.791	8.660.680	
Perturbing P_{RB}	40	0,05	0,3	0,9	17h 20min 25sec	300	8.996.096	8.344.285	
	40	0,05	0,7	0,9	7h 1min 4sec	121	8.951.530	8.640.283	
Perturbing P_c	40	0,05	0,5	0,99	20h 19min 39sec	300	8.973.727	8.346.790	
	40	0,05	0,5	0,8	17h 30min 34sec	300	8.933.736	8.639.942	

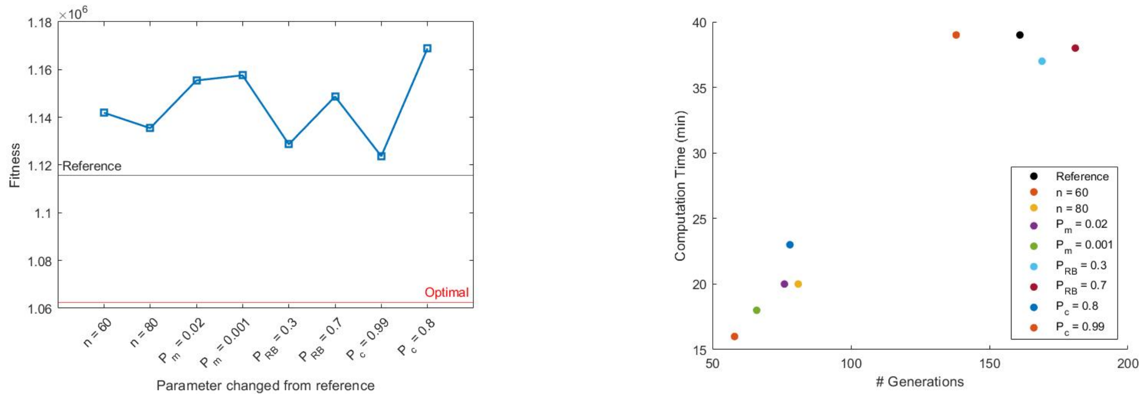
Table A.4: Results using the Genetic Algorithm with the modified mutation operator on Instance 4, testing different parameters.

Genetic Algorithm with time limit in mutation operator: **Instance 4**									
	Population size (n)	P_m	P_{RB}	P_c	Running Time	# Generations	Best Initial Solution	Best Final Solution	Best Integer Solution by Gurobi / Lower Bound
Reference	**40**	**0,05**	**0,5**	**0,9**	**11h 6min 35sec**	**119**	**9.003.036**	**8.643.809**	7.387.929 / 6.860.623
Perturbing n	60	0,05	0,5	0,9	5h 20min 20sec	74	8.912.811	8.650.503	
	80	0,05	0,5	0,9	7h 0min 5sec	101	8.953.718	8.589.062	
Perturbing P_m	40	0,02	0,5	0,9	9h 59min 10sec	122	8.969.059	8.619.595	
	40	0,001	0,5	0,9	4h 10min 55sec	177	8.964.325	8.695.071	
Perturbing P_{RB}	40	0,05	0,3	0,9	6h 57min 23sec	85	8.955.909	8.625.113	
	40	0,05	0,7	0,9	7h 3min 4sec	77	8.979.550	8.620.606	
Perturbing P_c	40	0,05	0,5	0,99	9h 53min 27sec	89	8.974.767	8.586.629	
	40	0,05	0,5	0,8	6h 57min 43sec	87	8.985.354	8.639.942	

Table A.5: Results using the Genetic Algorithm with time limit in the mutation operator on Instance 4, testing different parameters.

Genetic Algorithm with modified mutation operator: **Instance 5**									
	Population size (n)	P_m	P_{RB}	P_c	Running Time	# Generations	Best Initial Solution	Best Final Solution	Best Integer Solution by Gurobi
Reference	**40**	**0,05**	**0,5**	**0,9**	**11h 39min 56sec**	**120**	**12.868.423**	**12.544.873**	16.013.457
Perturbing n	60	0,05	0,5	0,9	12h 47min 51sec	120	12.872.681	12.522.208	
	80	0,05	0,5	0,9	10h 47min 31sec	120	13.016.689	12.455.481	
Perturbing P_m	40	0,02	0,5	0,9	8h 12min 39sec	91	12.884.598	12.583.770	
	40	0,001	0,5	0,9	6h 19min 22sec	60	12.866.932	12.690.036	
Perturbing P_{RB}	40	0,05	0,3	0,9	10h 47min 20sec	114	12.889.372	12.508.809	
	40	0,05	0,7	0,9	11h 48min 5sec	120	12.872.928	12.594.382	
Perturbing P_c	40	0,05	0,5	0,99	13h 1min 12sec	120	12.888.772	12.507.382	
	40	0,05	0,5	0,8	10h 54min 15sec	120	12.844.142	12.541.499	

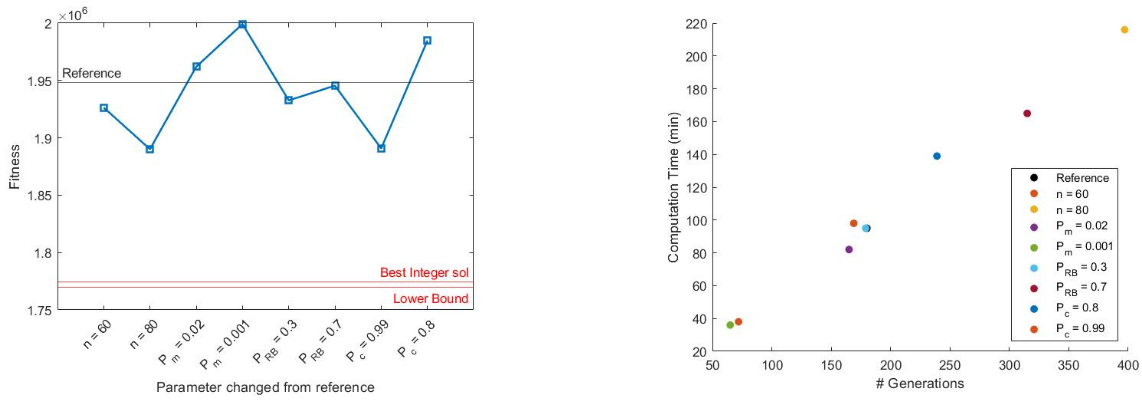
Table A.6: Results using the Genetic Algorithm with the modified mutation operator on Instance 5, testing different parameters.



(a) Fitness of final solutions

(b) Number of generations versus computing time

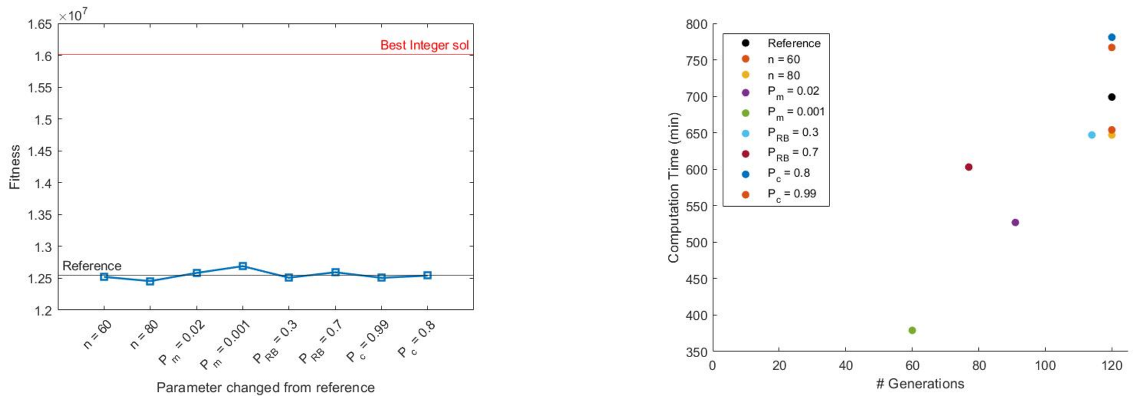
Figure A.1: Visual results of applying the GA to instance 1.



(a) Fitness of final solutions

(b) Number of generations versus computing time

Figure A.2: Visual results of applying the GA to instance 2.



(a) Fitness of final solutions

(b) Number of generations versus computing time

Figure A.3: Visual results of applying the GA to instance 5.

Bibliography

[1] Ruibin Bai, Edmund K. Burke, Graham Kendall, Jingpeng Li, and Barry McCollum. A hybrid evolutionary approach to the nurse rostering problem. *IEEE Transactions on Evolutionary Computation*, 14(4):580–590, 2010. ISSN 1089778X. doi: 10.1109/TEVC.2009.2033583.

[2] Jonathan F. Bard and Hadi W. Purnomo. A column generation-based approach to solve the preference scheduling problem for nurses with downgrading. *Socio-Economic Planning Sciences*, 39(3):193–213, 2005. ISSN 00380121. doi: 10.1016/j.seps.2004.04.001.

[3] X. Cai and K. N. Li. Genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operational Research*, 125(2):359–369, 2000. ISSN 03772217. doi: 10.1016/S0377-2217(99)00391-4.

[4] Philippe De Bruecker, Jorne Van Den Bergh, Jeroen Beliën, and Erik Demeulemeester. Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16, 2015. ISSN 03772217. doi: 10.1016/j.ejor.2014.10.038.

[5] Kenneth De Jong. Learning with Genetic Algorithms: An Overview. *Machine Learning*, 3(2):121–138, 1988. ISSN 15730565. doi: 10.1023/A:1022606120092.

[6] Gail W. DePuy, John S. Usher, Benjamin Arterburn, Randall Walker, and Maria Fredrick. Workforce training schedule for logistics skills. *2006 IIE Annual Conference and Exhibition*, 2006.

[7] John W. Fowler, Pornsarun Wirojanagud, and Esma S. Gel. Heuristics for workforce planning with worker differences. *European Journal of Operational Research*, 190(3): 724–740, 2008. ISSN 03772217. doi: 10.1016/j.ejor.2007.06.038.

[8] Daniel Grieshaber. ThinkIR : The University of Louisville ' s Institutional Repository A hybrid genetic-greedy approach to the skills management problem . By Department of Computer Engineering and Computer Science. 2009.

[9] Sin C. Ho and Janny M.Y. Leung. Solving a manpower scheduling problem for airline catering using metaheuristics. *European Journal of Operational Research*, 202(3):903–921, 2010. ISSN 03772217. doi: 10.1016/j.ejor.2009.06.030. URL `http://dx.doi.org/10.1016/j.ejor.2009.06.030`.

[10] Huei Chuen Huang, Loo Hay Lee, Haiqing Song, and Brian Thomas Eck. SimMan-A simulation model for workforce capacity planning. *Computers and Operations Research*, 36(8):2490–2497, 2009. ISSN 03050548. doi: 10.1016/j.cor.2008.10.003.

[11] Erin Jackson. ThinkIR : The University of Louisville ' s Institutional Repository Skills management heuristics. 2009.

[12] Ninghua Li and Ling X. Li. Modeling staffing flexibility: A case of China. *European Journal of Operational Research*, 124(2):255–266, 2000. ISSN 03772217. doi: 10.1016/S0377-2217(99)00379-3.

[13] Serpil Sayin and Selçuk Karabati. Assigning cross-trained workers to departments: A two-stage optimization model to maximize utility and skill improvement. *European Journal of Operational Research*, 176(3):1643–1658, 2007. ISSN 03772217. doi: 10.1016/j.ejor.2005.10.045.

[14] Pieter Smet, Burak Bilgin, Patrick De Causmaecker, and Greet Vanden Berghe. Modelling and evaluation issues in nurse rostering. *Annals of Operations Research*, 218(1):303–326, 2014. ISSN 15729338. doi: 10.1007/s10479-012-1116-3.

[15] Swati Swayamsiddha. Bio-inspired algorithms: principles, implementation, and applications to wireless communication. In Yang Xin-She, editor, *Nature-Inspired Computation and Swarm Intelligence*, chapter 4, pages 49–63. Elsevier Ltd, 2020. ISBN 9780128197141. doi: 10.1016/b978-0-12-819714-1.00013-0. URL `https://doi.org/10.1016/B978-0-12-819714-1.00013-0`.

[16] Vikram Tiwari, James H. Patterson, and Vincent A. Mabert. Scheduling projects with heterogeneous resources to meet time and quality objectives. *European Journal of Operational Research*, 193(3):780–790, 2009. ISSN 03772217. doi: 10.1016/j.ejor.2007.11.005. URL `http://dx.doi.org/10.1016/j.ejor.2007.11.005`.

[17] Rodney B. Wallace and Ward Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management*, 7(4):276–294, 2005. ISSN 15234614. doi: 10.1287/msom.1050.0086.

[18] Haibo Wang, Bahram Alidaee, Jaime Ortiz, and Wei Wang. The multi-skilled multi-period workforce assignment problem. *International Journal of Production Research*, 59(18):5477–5494, 2021. ISSN 1366588X. doi: 10.1080/00207543.2020.1783009. URL `https://doi.org/10.1080/00207543.2020.1783009`.

[19] Yang Xin-She. Genetic algorithms. In *Nature-Inspired Optimization Algorithms*, chapter 6, pages 91–100. Second edition, 2021. ISBN 9780128219867. doi: 10.1016/B978-0-12-821986-7.00013-5.