INTERDEPARTMENTAL MASTER PROGRAM

IN BUSINESS ADMINISTRATION

Diploma Thesis

"Optimization of operating room efficiency"

Giannakopoulos Ioannis

In fulfillment of the requirements for the Diploma of Master in Business Administration.

December 2020

ABSTRACT

The organized provision of medical care to individuals or a community is a must in our societies. Hospitals, clinics, and medical centers are actively trying to solve the problems they face every day.

Most of the problems are focused on management and organization. Especially in areas where the population is big and the medical cases are multiple, the patients' flow is critical for their quality.

Each health organization is trying to reduce the cost of the services, but at the same time is trying to save lives or at least offer a good quality of life to their patients. Thousands of people need daily medical help, and most of the time, medical cases are critical and urgent.

The main problem of the healthcare system is that the available resources are limited. So, the optimal usage of the available resources is the only solution. If someone tries to do this planning manually to find the optimal solution, they will probably need a lot of time, and finally, the result might not be optimal. A wrong decision of resource usage will cost money, delays, dissatisfaction, and the worst, in many cases, lives will get lost because of bad management.

Of course, a human cannot predict or calculate all the variables and the parameters, especially in complicated problems. Linear programming is a viable solution because it helps to find the optimal solution efficiently and accurately, depending on the complexity of the problem.

The operating room (OR) is for all healthcare facilities one of the most critical resources. The correct usage of an operating room affects the cost and the revenues of the medical center and handles very serious and important human health cases. Although the operating room is one of the most important and dynamic hospital departments, it is, unfortunately, one of the departments that presents problems in the organization and planning.

Each operating room is equipped with special tools, has special environmental conditions, and that is why each facility has separate rooms just for surgeries. The number of operating rooms (ORs) per hospital is limited, but the cases come from different departments. Except for the operating rooms and the equipment, each hospital has specialized staff, like doctors and nurses, for surgeries. This is the reason that the surgical teams are limited too.

In this thesis, we describe a way to optimize the operating room allocation by formulating a linear programming model, using python (programming language).

ACKNOWLEDGEMENTS

There are some people I would like to thank for their patience and support.

One of them is Associate Professor Loukas Tsironis for his beneficial guidance, sharing useful research sources, professional attitude, and friendly approach.

Also, my wife and my two daughters, for their understanding and their patience.

CONTENTS

TABLES

FIGURES

CHAPTER 1

Introduction

History

There is a mystery behind the large number of trepanned skulls found worldwide since Neolithic times. It seems that these were the first attempts at primitive cranial surgery. More than 1500 trepanned skulls have been discovered worldwide, from Europe and Scandinavia to North America, from Russia and China to South America (especially in Peru).

When the skulls are examined, there is evidence of recovery in some cases, and we suspect that the patient survived. But in many cases, the surgery seems to have been incomplete. Perhaps the patient suddenly woke up and abandoned the operation.

Therefore, the science of surgery seems to have begun long ago, although there is no information as to when. Perhaps then, and for a long time before that, the motivation to perform surgery were not the same as they are today. We can assume that medicine men performed these cranial operations. However, we cannot deny that an important reason for them to perform surgery may have been to modify human behavior - a specialty called psychosurgery in the mid-20th century!

There are also traces of attempts to anesthetize a patient in the writings of the Sumerians, Babylonians, Assyrians, Egyptians, Indians, and Chinese. Over the years, anesthesia has made significant advances in science and medicine.

In the last two centuries, surgery has become radically more effective, and its violence greatly reduced. All these changes have proved central to the development of man's ability to heal the sick.

Progress on surgical science was minimal until about 1800. Most accounts focused on medicine and general medical progress. From the mid-1800s to the mid-1900s, there was an evolution in terms of surgical science advancement. Many articles were published describing the development in this field.

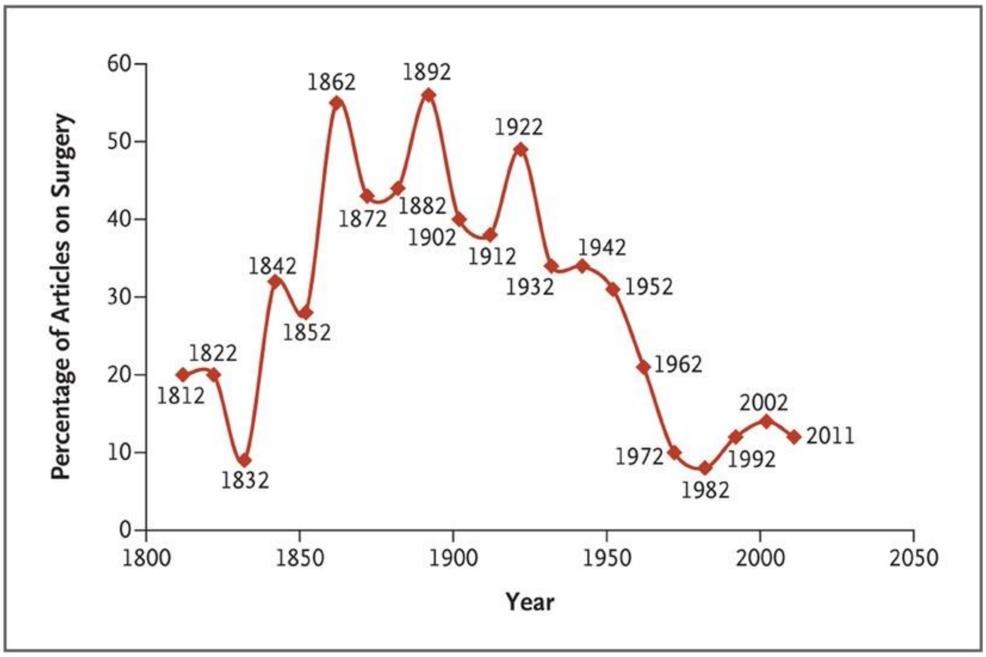
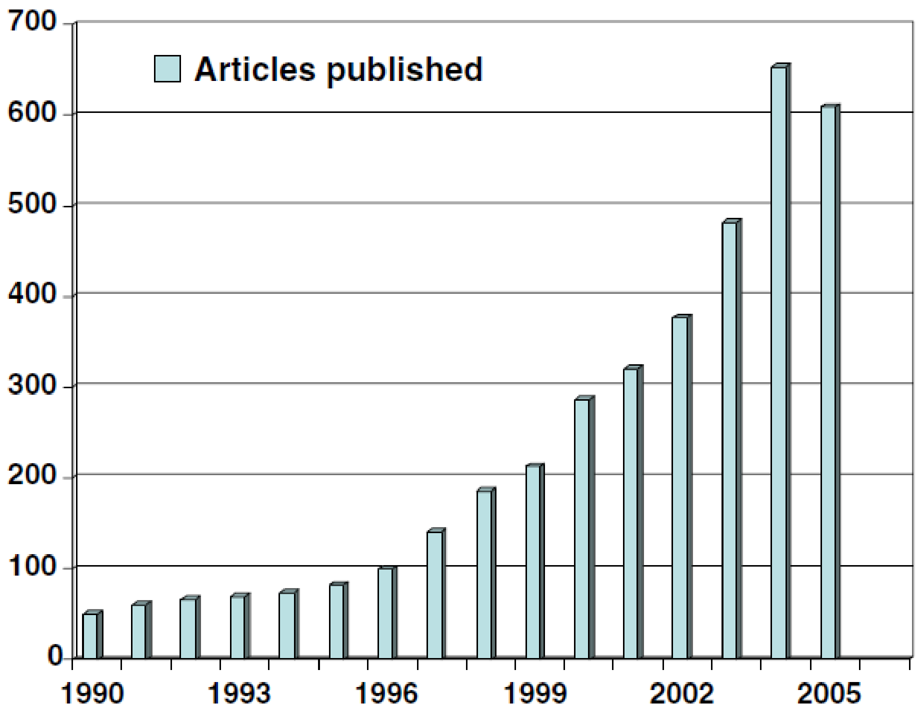


Figure 0.1 Changes in the Proportion of Articles on Surgery Published in Journal since 1812[1]

[1] (Gawande, 2012)

Evolution, technological development, and acquired knowledge made surgical science one of the most important sciences in medicine. However, the specialization needed in terms of the knowledge and the tools classifies it as one of the most expensive sciences.

In the beginning of the 20th century, a vital technological tool appeared to catalyze humankind's progress – robots. They increased the rate of progress, and especially in surgical science, robotics' science gave a significant push. The concept of robotics penetrated the general consciousness. They helped in improving services, creating new techniques, reducing the need to perform surgery and made healing and recovery faster. In surgical science however, besides the knowledge required, the specialization in the application of this technology is also necessary.



Number of articles indexed each year in a pubmed search containing the words "robot" or robotic.

Figure 0.2 Graph depicting the number of manuscripts indexed per year in a PubMed search containing the words "robot" or "robotic"[2]

[2] (Hockstein, 2007)

The growing population, and therefore the patients who need to be operated on and cared for, make it necessary to manage operating rooms appropriately and efficiently.

Alex Macario (A., 2006), the author of the article, “Are Your Hospital Operating Rooms "Efficient"? Scoring System with Eight Performance Indicators” shows the next table called "A Scoring System for OR Efficiency" where we can see the eight performance indicators and each one's weight for OR efficiency. Although surgeon satisfaction is also critical, no valid and reliable tool has yet been developed to measure this.

Table 1. A Scoring System for OR Efficiency

	Points		
Metric	0	1	2
Excess staffing costs	Greater than 10%	5–10%	Less than 5%
Start-time tardiness (mean tardiness of start times for elective cases per OR per day)	Greater than 60 min	45–60 min	Less than 45 min
Case cancellation rate	Greater than 10%	5–10%	Less than 5%
PACU admission delays (% of workdays with at least one delay in PACU admission)	Greater than 20%	10–20%	Less than 10%
Contribution margin (mean) per OR hour	Less than $1,000/h	$1,000–2,000/h	More than $2,000/h
Turnover times (mean setup and cleanup turnover times for all cases)	Greater than 40 min	25–40 min	Less than 25 min
Prediction bias (bias in case duration estimates per 8 h of OR time)	Greater than 15 min	5–15 min	Less than 5 min
Prolonged turnovers (% of turnovers that are more than 60 min)	Greater than 25%	10–25%	Less than 10%

OR = operating room; PACU = postanesthesia care unit.

Figure 0.3 A Scoring System for OR Efficiency

Operating Room Management

OR management has become the science of planning and increasing the efficiency of OR utilization in a healthcare facility. It represents a healthcare facility's ability to manage their patients' health and safety and combine their financial data. Besides everything else, a good and smooth OR schedule is also a matter of marketing for the facility.

There are two classes of patients considered in OR scheduling and categorization, namely elective and non-elective patients. In the first category, the patients are those for whom the surgery can be planned, and in the category of non-elective patients, the surgery is unexpected and needs to be done urgently.

There are several criteria to evaluate the performance of surgery scheduling. One important one is patient or surgeon waiting time. Peter VanBerkel and John T.Blake (Peter VanBerkel, 2012) mention that waiting time is related to throughput analysis. They influence the throughput by changing the capacity of beds and the time when operating rooms are available.

F. Dexter (Dexter., 2000) approaches the scheduling problem based on operating room efficiency. He measures the underutilization and overutilization of the operating room. Each department arranges surgeries and based on the number of expected cases, each day requires a certain number of operating rooms. In a hospital, the total number of operation theaters is limited, so there must be a schedule for optimal allocation to each department, depending on its needs. In this way, each department knows how many hours of operation it has available per day and organizes the upcoming operations accordingly. As E. Marcon and F. Dexter (Marcon & Dexter, 2007) simulate less detailed parameters, such as longest case first or shortest case first, this can help reduce the peak number of patients both in the waiting area and in the post anesthesia care unit (PACU). As they mention in their study about the results they had: "The surgeons differed significantly among themselves in their sequencing of cases and were also internally nonsystematic, based on case durations. The functional effect of this uncoordinated sequencing was for the surgical suite to behave overall as if there was random sequencing. The resulting PACU staffing requirements were the same as those of the best sequencing method identified in prior simulation studies. Although sequencing "Longest Cases First" performs poorly when all ORs have close to 8 h of cases, at the studied hospital it performed no worse than the other methods. The reason was that some ORs were much busier than others on the same day. The standard deviation among ORs in the hours of cases, including turnovers, was 3.2 h: largely relative to the mean workload. Data from 33 other hospitals confirmed that this situation is commonplace. Additional studies showed that case sequencing also had minimal effects on the peak number of patients in the holding area."

Perhaps this approach to scheduling seems simplistic, but as we have seen (at least in those cases where their method performed poorly), the results were no worse than other methods. However, a complicated program that considers a large number of parameters, variables, and data to be computed takes a long time to find a solution and is ultimately not suitable for a large number of cases.

The departments in an organization are dynamic in nature, and day by day, the data changes. Especially in a clinic, the staff, the medical cases, the equipment needed, and each case's duration are different every day. Therefore, it is challenging to organize many clinics simultaneously as hospital management has to do to accommodate all department cases.

Many OR scheduling approaches reduce the number of canceled elective surgeries, target cost, or stakeholder preferences. This paper focuses on OR capacity planning rather than OR management or issues such as business process reengineering, new medical technologies that might affect OR duration or facility design.

Another approach to the problem, similar to this paper's philosophy, is Hillier and Lieberman's method (Hillier, 2004). They solve it in 2 phases. In the first phase, there are the operations (m) to be performed in (n) operating rooms. Another parameter that is inserted is the physician's preference (p_{ij}) to operate (i) in an operating room (j). Each surgery requires a certain time, and there is an available time constraint for each operating room. In the first phase, the program maximizes the doctor's preferences for operating rooms, and in the second phase, the operations are ranked.

In the future, it seems, operating rooms will face the same scheduling problem. Even though the development of technology to telecommunication, intraoperative imaging (real-time or three-dimensional), robotics will likely change the time of an operation, but not the cost to the operating room and of course, scheduling will continue to be an issue because of the complexity of the problem.

CHAPTER 2

Subject of examination

The Papageorgiou hospital

Each medical care facility has a certain number of operating rooms to handle all its clinics' medical cases. For this thesis, we have taken the "Papageorgiou" Hospital, in Thessaloniki, city of Greece. Papageorgiou Hospital is one of Greece's largest hospitals, and its technical equipment places it among the top twenty hospitals in the world. Papageorgiou Thessaloniki General Hospital is a Legal Entity at Private Law, a non-profit that provides prevention, diagnosis, treatment, and rehabilitation services.

Figure 0.1 General Hospital Papageorgiou Thessaloniki (Arial View)

Started in August 1999, the establishment of University Clinics was completed in 2004 when the hospital was staffed, staffing is now in constant collaboration with the Medical School of the Aristotle University of Thessaloniki.

After more than ten years of operation, and according to an evaluation carried out by an international committee on behalf of AUTh, Papageorgiou Hospital is considered one of the best care facilities in Europe in terms of completeness and quality of the services provided to patients.

The number of employees is 1502, the number of beds is 800, and more than 1000 people come to the Outpatient Clinics daily. The number of cases treated at the Emergency Department exceeds 900 on average, and when students, attendants, and visitors are added, the daily traffic on the hospital grounds exceeds 7000. The annual number of operations is about 18,264, which means that about 50 to 70 operations are performed every day. The operating rooms (ORs) of the hospital are:

1st: Pediatric Surgery

2nd & 3rd: General Surgery

4th: Ophthalmology

5th: Neurosurgery

6th & 7th: Thoracic Surgery and Cardiac Surgery

8th & 9th: Orthopedics

10th: University Surgery

11th: Otolaryngology (ENT).

12th: Plastic Surgery

13th: Urology

14th: Septic Surgery,

The types of surgeries that take place are Open Surgery, Laparoscopic Surgery, Vascular Surgery, Pediatric Surgery, Neonatal Surgery, Ophthalmology, Cardiac Surgery, Thoracic Surgery, Orthopedics, Trauma Surgery, Orthopedic Surgery, Plastic Surgery, Urological Surgeries - Open as well as Diuretic Surgeries and ENT surgeries.

From the information available, we can understand there are various types of surgeries, a lot of clinics, and a large number of patients to serve daily. Obviously, if someone tries to manually schedule all surgeries in the available operating rooms, considering the available surgery teams per day, it would probably be complicated to find the optimal solution to meet all parameters.

Below we can see how the operating room of Papageorgioy hospital looks like and how it is equipped.

Figure 0.2 Operating Room (view 1)[3]

[3] Source: https://www.papageorgiou-hospital.gr/

Figure 0.3 Operating Room (View 2)[4]

The Clinics

Each hospital department is independently organized and operates autonomously, and the management must combine the data from each clinic and satisfy them fairly. We have to mention each clinic's needs because some surgeries are taking place only for some hours or for a couple of days per week. Considering these needs, we are going to build the model.

One of the most important clinics of the hospital is Heart-Thoracic-Vascular Surgery Clinic. Till now, at least 2000 cardiac surgeries have already been performed considering this clinic.

700 thoracic operations.

150 major thoracic, vascular surgeries and 500 lesser severity surgeries.

The Neurosurgery Clinic is another important clinic. The clinic is staffed by 6 specialized neurosurgeons who cover the broadest spectrum of operations in modern Neurosurgery Specialty.

---

[4] Source: https://www.papageorgiou-hospital.gr/

The clinic performs Morning Regular Outpatient Clinic general neurosurgery daily, and clinicians perform Afternoon Outpatient Clinics three times a week.

Another clinic is the Plastic Surgery Clinic (AUTh). The main goal and sole object of Plastic, Reconstructive, and Esthetic Surgery are to restore the organic functionality of the citizen in harmony with himself and his social environment.

Normally, the number and frequency of operations for the plastic surgery clinic are :

One (1) room of regular general operations three (3) times per week.

One (1) room of Clinical Short Care Hospital 2 times per week

One room of minor surgery (3 times per week).

The Pediatric Surgery is another important clinic, with specializations in Neonatal Surgery, Laparoscopic Surgery, Thoracic Surgery (VATS), Endoscopies, Diagnostic biopsies of all systems and some new specializations in Pediatric Surgery such as Robotic Surgery; the Application of Laser Beams in Pediatric Surgery, the wider application of Thoracic Surgery (VATS).

The outpatient clinics of Pediatric Surgery are in general as shown below:

- Regular Outpatient Clinics (ROC): are held twice a week (Tuesday and Thursday) from 09:00 to 13:00 (2 times per week X 4 hours = 8 hours/week) for cases of General Pediatric Surgery.

- Special ROC of Neonatal Surgery every Wednesday 12:00-13:00 (1 hour)

- Special ROC for Craniofacial Diseases and Clefts every Friday 12:00-13:00 (1 hour)

- Especially ROC Congenital Protests of the Urinary Tract System: every Monday 12:00-13:00 (1 hour)

- Emergency Department (ED) In the clinic's on-call hours, which amount to 6-7 times a month.

So, this department requires 11 – 12 hours per week on specific days.

The First Surgery Clinic operates regular outpatient clinics five days a week:

• General surgery five times a week (Monday to Friday)

• Clinic of upper digestive diseases twice a week (Wednesday and Friday)

• Clinic for lower digestive diseases every Wednesday and Thursday

• Advanced Laparoscopic Surgery Clinic every Tuesday

• Breast clinic every Monday and Endocrine Gland Clinic every Wednesday

This means that the clinic needs, depending on the day of the week, at least one vacant hospital operating room as a minimum requirement.

Additionally, there is the First University Surgery Clinic. More than 200 medical students are trained per year in this clinic in the compulsory course of General Surgery, in the 6th, 7th, and 11th semesters. The daily clinic operating schedule is:

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
OPERATIONS GEN. SURGERY	•	•	•	•	•
VASCULAR SURGERIES	•	•	•	•	•
SHORT CARE			•		
SEPTIC SURGERIES				•	

Table 0.1 General Surgery Clinic Weekly Schedule

All the above information from the hospital gives the planner some indication of each clinic's needs per week and the goals, more or less, of each department.

CHAPTER 3

The system under consideration

The Clinics' requirements

Gathering the requirements of each clinic, we have the following information using which a model can be constructed:

- There are 14 operating rooms in the hospital to service the existed clinics.

- The number of the clinics that need scheduling are 11 considering Pediatric Surgery, General Surgery, Ophthalmologist, Neurosurgery, Thoracic Surgery, and Cardiac Surgery, Orthopedics, University Surgery, Otolaryngology, Plastic Surgery, Urology, and Septic Surgery.

- The scheduled use of the operating rooms will be for 7 hours per day or less because of unexpected delays.

- Neurosurgery Clinic has available 6 surgical teams

- Plastic Surgery Clinic needs 8 operating rooms per week as mentioned before (One (1) room of the Regular General Surgeries three (3) times a week, One (1) room of the Clinical Short Care Hospital 2 times a week, Minor surgery room (3 times a week)).

- Pediatric Surgery needs operating rooms during the week for about 11-12 hours. We can consider that the clinic's needs are about 1 operating room, twice a week.

The Papageorgiou hospital records every patient who needs surgery. One important piece of information that the hospital keeps in view is the deadline that the doctor sets for the surgery to be done. Every clinic checks the flow and the average number of medical cases, asking for the available operating rooms' capacity.

These are each clinic's current needs, and consequently, the minimum targets that they try to achieve. With the optimization program, the hospital management will make different scenarios get the optimal solution if the needs for a clinic changed.

The operating room planning problem is complicated with many variations, so some assumptions were made to define the problem under consideration:

1. All the available operating rooms are multifunctional. That means that each surgical case can be operated on in any of the available operating rooms.

2. No surgeon has priority to reserve a block of time in advance, so none of the surgeons can prioritize the operating room capacity given from the program.

3. Human and instrumental resources, except for surgeons, are always available whenever they are needed. So, each surgeon has their surgery team always available.

4. There are always sufficient recovery beds. If surgery is finished, there is always a recovery bed available for the patient to recover. If not, the surgery is canceled, and the department's needs are reduced from the very beginning.

5. Emergency cases are not taken into consideration. Such cases are accommodated in the daily operating hours i.e., 7.

6. Once a surgery begins in an operating room, it cannot be interrupted.

It is critical for the hospital to wisely allocate the available resources, as we mentioned before. For our case (Papageorgiou hospital), we can create a table with the 11 clinics and the operating room capacity that these clinics require per week. This suggests that more than 14 operating rooms are required per day, this exceeds the available operating rooms. Below we can see a visual representation:

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Pediatric surgery	**1**	**2**	**1**	**2**	**1**
General Surgery	**4**	**4**	**8**	**4**	**4**
Ophthalmologist	**2**	**2**	**2**	**4**	**2**
Neurosurgery	**2**	**2**	**2**	**2**	**2**
Thoracic surgery and Cardiac surgery	**4**	**4**	**4**	**4**	**4**
Orthopedics	**2**	**2**	**2**	**2**	**2**
University Surgery	**2**	**2**	**2**	**2**	**4**
Otolaryngology	**2**	**1**	**2**	**1**	**2**
Plastic Surgery	**4**	**2**	**4**	**2**	**4**
Urology	**2**	**2**	**2**	**2**	**2**
Septic Surgery	**1**	**1**	**1**	**1**	**2**

Table 0.1 Number of Surgical Teams per Department per Day

And if we sum each column:

	Monday	Tuesday	Wednesday	Thursday	Friday
Daily Demands	**26**	**24**	**30**	**26**	**29**

Table 0.2 Daily sum of operating rooms demands

Since the available operating rooms are 14 per day, this thesis attempts to create a linear programming model to maximize allocation of operating rooms to clinics. The first constraint of our problem is that the allocation cannot exceed 14 rooms per clinic:

$$\sum_{i=1}^{11} x_{ij} \leq 14 \quad j=1,\ldots,5.$$

Formulating a Model

In this chapter the parameters of the problem, the variables, and the constraints have been discussed. These will be used to formulate the problem, the solution of which has been discussed in the next chapter.

If one clinic estimates that it needs one functional operating room on any given day, it will also require at least one surgical team for these 7 hours of the current day. We set 7 hours per day and not 8 because we assume that about 1 hour daily is the non-working hour (e.g., the time between surgeries).

All the needed data will be represented as tables in an Excel file, from where the program will read. In the end, the program will share the optimal allocation of ORs for each department by creating a new table showing a one-week schedule. The **objective function** to maximize the sum of the percentages of the weekly targets achieved is given below:

$$max\, \mathrm{Z} = \sum_{i=1}^{11} \frac{7}{t_i} \sum_{j=1}^{5} x_{ij}$$

To solve the problem, we proceed with the understanding that one surgical team is available for each operating room. Therefore, we can use table 3.1.1 to obtain the number of the available surgical teams (a_{ij}), where “i” is the department and “j” represents days of the week. This means that the optimal solution for the allocation of operating rooms (x_{ij}) cannot be more than the available surgical teams in each department. This forms the second constraint in our problem:

$$x_{ij} \leq a_{ij} \quad i=1,\dots,11 \quad j = 1,\dots,5.$$

Each clinic is therefore constrained not only by the number of available surgical teams (a_{ij}) for each day of the week, but also by the number of rooms. However, some surgeries must be done due to their criticality, so each department sets the minimum requirement of surgical teams and ORs. To account for the minimum requirement, we create a similar table with the minimum operating rooms per department per day (m_{ij}^{-}) (see table 3.2.2). Since Pediatric Surgery needs a maximum of 11 – 12 hours weekly, the minimum requirement is set to zero for this clinic.

Min number of ORs per day	Monday	Tuesday	Wednesday	Thursday	Friday
Pediatric surgery	**0**	**0**	**0**	**0**	**0**
General Surgery	**2**	**2**	**4**	**2**	**2**
Ophthalmologist	**0**	**0**	**0**	**2**	**0**
Neurosurgery	**1**	**1**	**1**	**1**	**1**
Thoracic surgery and Cardiac surgery	**2**	**2**	**2**	**2**	**2**
Orthopedics	**2**	**2**	**1**	**2**	**1**
University Surgery	**1**	**1**	**0**	**1**	**2**
Otolaryngology	**1**	**0**	**1**	**0**	**0**
Plastic Surgery	**2**	**1**	**2**	**1**	**2**
Urology	**0**	**0**	**0**	**0**	**1**
Septic Surgery	**0**	**0**	**0**	**0**	**0**

Table 0.1 Minimum number of ORs per day

Additionally, a table gives information about the maximum number of operating rooms required by each department (m_{ij}^{+}). This represents the maximum number of operating rooms that each clinic can handle per day. It has the same format as table 3.2.1. It can be presented as:

$$m_{ij}^{-} \leq x_{ij} \leq m_{ij}^{+} \quad i=1,\dots,11 \quad j = 1,\dots,5.$$

Therefore, if the daily number of operating rooms are bound between the minimum and the maximum requirements, the weekly requirements can be deduced:

$$w_i^- \leq \sum_{j=1}^{5} x_{ij} \leq w_i^+ \quad i=1,\dots,11.$$

Where (w_i^-) and (w_i^+) are the minimum and maximum respectively, weekly requirements for operating rooms in department *i*.

Max number of ORs per day	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Pediatric surgery	**2**	**2**	**2**	**2**	**2**
General Surgery	**8**	**8**	**8**	**8**	**8**
Ophthalmologist	**2**	**2**	**2**	**2**	**2**
Neurosurgery	**2**	**2**	**2**	**2**	**2**
Thoracic surgery and Cardiac surgery	**4**	**4**	**4**	**4**	**4**
Orthopedics	**3**	**3**	**3**	**3**	**3**
University Surgery	**4**	**4**	**4**	**4**	**4**
Otolaryngology	**2**	**2**	**2**	**2**	**2**
Plastic Surgery	**4**	**4**	**4**	**4**	**4**
Urology	**2**	**2**	**2**	**2**	**2**
Septic Surgery	**1**	**1**	**1**	**1**	**1**

Table 0.2 Maximum number of ORs per day

If we check and compare the three tables, we can see the differences between them:

	Weekly Requirements	Min. Weekly Requirements	Max. Weekly Requirements
Pediatric surgery	**7**	**0**	**10**
General Surgery	**24**	**12**	**40**
Ophthalmologist	**12**	**2**	**20**
Neurosurgery	**10**	**5**	**10**
Thoracic surgery and Cardiac surgery	**20**	**10**	**20**
Orthopedics	**10**	**8**	**10**
University Surgery	**12**	**5**	**20**
Otolaryngology	**8**	**2**	**10**
Plastic Surgery	**16**	**8**	**20**
Urology	**10**	**1**	**10**
Septic Surgery	**6**	**0**	**6**
Total	**135**	**53**	**176**

Table 0.3 Requirement's compare tables

The next step is to insert the weekly targets per department (t_i). We can set each clinic's weekly operating hours as a target. The optimal solution would be to use the available Surgical Teams per day for at least 7 hours. This parameter is a percentage of the available weekly requirements (in hours) that each department sets as the target to achieve. So, if two clinics have the same number of surgical teams available and consequently the same weekly hours, one of the clinics requirements can be satisfied if it reaches 75% of these hours while the other could be satisfied at 60%. Therefore, the program will assign different weights to each clinic. For the Pediatric surgery, we set the target of 14 hours, and for the septic surgery, only 7.

The weekly targets per department are shown below:

	Weekly Targets
Pediatric surgery	**14**
General Surgery	**126**
Ophthalmologist	**50.4**
Neurosurgery	**49**
Thoracic surgery and Cardiac surgery	**105**
Orthopedics	**56**
University Surgery	**54.6**
Otolaryngology	**28**
Plastic Surgery	**56**
Urology	**35**
Septic Surgery	**7**

Table 0.4 Weekly Targets per Department

The solution of this problem will achieve the highest percentage of the weekly targets for each department, considering the available operating rooms of the hospital. The program will distribute the available resources over all the clinics on a daily basis, trying to meet each department's requirements as best as possible.

The code to solve the problem was written in Python 3, in the form of a Jupyter notebook, in the Anaconda development environment. The development took place gradually in this environment, in parallel with the tests for the results' correctness.

In the next chapter, we will give some information about Python, Jupyter, Anaconda, GitHub, and Binder, and we are going to explain how the program is created, how the code was written, and the results we get after running the program.

CHAPTER 4

The Mathematical Model and the Programming Code

4.1 The Linear Programming - Problem formulation

Before we begin with the solution, lets sum up the developments from the previous chapters. In this paper, we plan to optimally allocate operating rooms to departments of a hospital so as to achieve the highest possible fraction of the hours demanded by the departments for operating rooms.

This is an integer programming (IP) problem. The mathematical formulation is as follows. The decision variables (x_{ij}) takes integer values to denote number of operational rooms assigned to department *i*, on the day *j*. Such that, $x_{ij} \in \mathbb{N}$, where i $\in [1,11]$ and j $\in [1,5]$.

The **objective** is to **maximize** the sum of operational rooms allocated to hospital departments to achieve weekly targets. For each department *i*, the total hours worked per week are $7\sum_{j=1}^{5} x_{ij}$. If t_i is the hours requested from the department *i* that were fulfilled in a week, then the fraction of the weekly target fulfilled is given by $\frac{7}{t_i}\sum_{j=1}^{5} x_{ij}$.

Therefore, **the objective function is**:

$$\max Z = \sum_{i=1}^{\mathbf{11}} \frac{7}{t_i} \sum_{j=1}^{5} x_{ij}$$

The constraints are as follows:

The total number of operating rooms each day has to be less than or equal to 14:

$$\sum_{i=1}^{11} x_{ij} \leq 14 \quad j=1,\dots 5 \qquad (C_1)$$

The operating rooms are also a) greater than or equal to the minimum specified ones, for each department and day, b) less than or equal to the maximum specified ones, for each department and day:

$x_{ij} \geq m_{ij}^{-}$ *i=1,…,11* *j = 1,…,5* *(C₃)*

$x_{ij} \leq m_{ij}^{+}$ *i=1,…,11* *j = 1,…,5* *(C₄)*

where m_{ij}^{-} and m_{ij}^{+} are the minimum and maximum values, per department *i* and day *j*.

In each department, the weekly numbers of operating rooms are less than or equal to the weekly targets:

$\sum_{j=1}^{5} x_{ij} \leq t_i$ *i=1,…,11* *(C₇)*

where are the weekly targets in department i.

The **decision variables** are:

x_{ij} nonnegative integers: $x_{ij} \in Z^{*} = \{0\} \cup Z^{+}$ *i=1,…,11* *j = 1,…,5*

While formulating the problem, **the following assumptions were made**:

- All the available operating rooms are multifunctional. That means that each surgical case can be operated on in any of the available operating rooms.
- No surgeon has priority to reserve a block of time in advance, so none of the surgeons can prioritize the operating room capacity given from the program.
- Human and instrumental resources, except for surgeons, are always available whenever they are needed. So, each surgeon has their surgery team always available.
- There are always sufficient recovery beds. If surgery is finished, there is always a recovery bed available for the patient to recover. If not, the surgery is canceled, and the department's needs are reduced from the very beginning.
- Emergency cases are not taken into consideration. Such cases are accommodated in the daily operating hours i.e., 7.
- Once a surgery begins in an operating room, it cannot be interrupted.

- There are 8 hours per day out of which 1 hour is a nonworking hour, i.e., the time between surgeries.

4.2 Solving the Problem

The proposed model was solved on Python 3.0 using the MIP (Mixed Integer Programming) package. The model was directed to 'Maximize' the Integer Programming problem. Initially the model is parametrized to select a strategy which gives priority to finding feasible solutions quickly and then improving on them, rather than moving directly towards an optimal solution. 55 decisions variables were adjusted to maximize the allocation of hours to all departments in 9.5 seconds to give the following optimal solution.

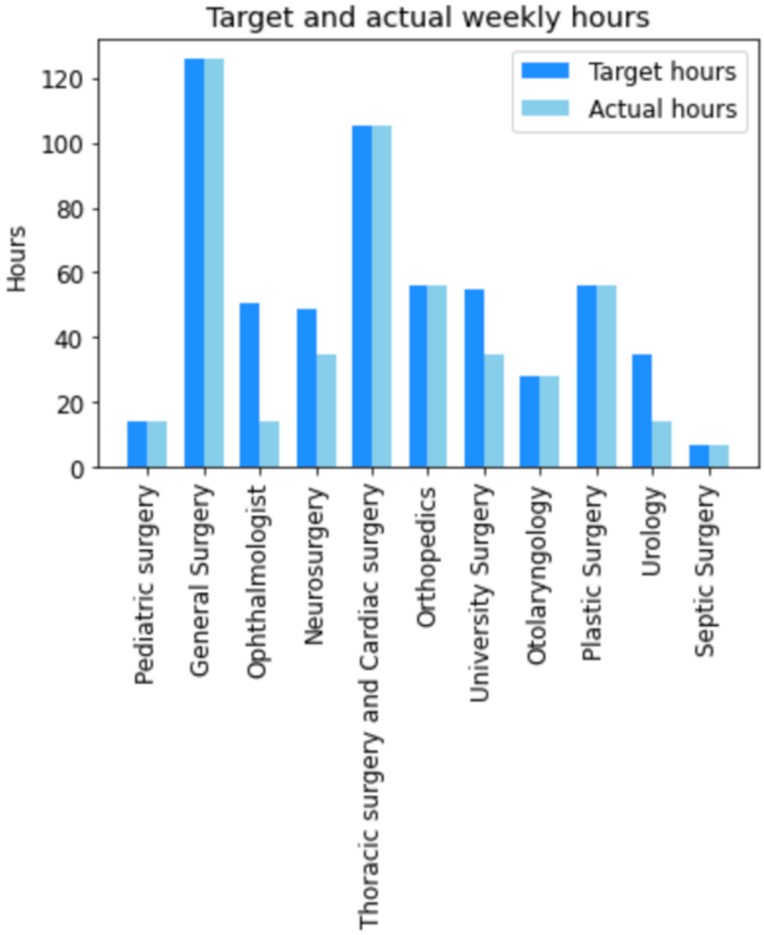
```
optimal solution cost   9.03 found in      1.9 seconds
solution:
x1_1 :   0
x1_2 :   1
x1_3 :   0
x1_4 :   1
x1_5 :   0
x2_1 :   3
x2_2 :   3
x2_3 :   6
x2_4 :   3
x2_5 :   3
x3_1 :   0
x3_2 :   0
x3_3 :   0
x3_4 :   2
x3_5 :   0
x4_1 :   1
x4_2 :   1
x4_3 :   1
x4_4 :   1
x4_5 :   1
x5_1 :   3
x5_2 :   3
x5_3 :   3
x5_4 :   3
x5_5 :   3
x6_1 :   2
x6_2 :   2
x6_3 :   1
x6_4 :   2
x6_5 :   1
x7_1 :   1
x7_2 :   1
x7_3 :   0
x7_4 :   1
```

```
x7_5 :    2
x8_1 :    1
x8_2 :    1
x8_3 :    1
x8_4 :    0
x8_5 :    1
x9_1 :    2
x9_2 :    1
x9_3 :    2
x9_4 :    1
x9_5 :    2
x10_1:    1
x10_2:    0
x10_3:    0
x10_4:    0
x10_5:    1
x11_1:    0
x11_2:    1
x11_3:    0
x11_4:    0
x11_5:    0
```

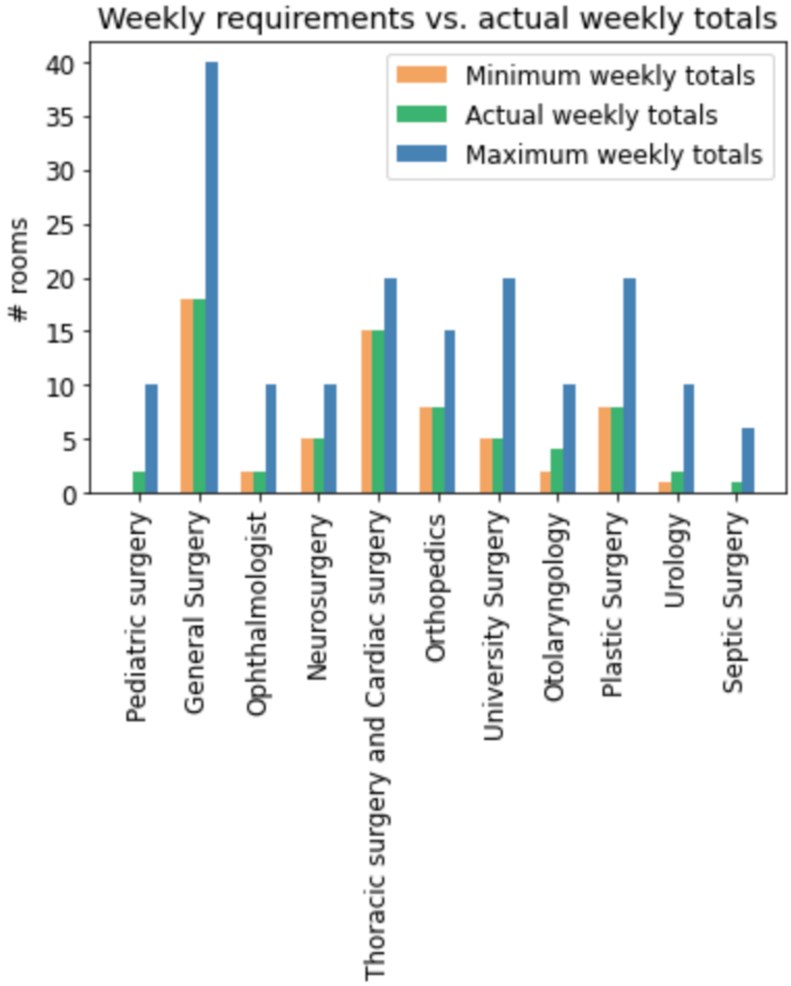
The above solution proposes the following allocation of operation hours to the departments. The cost of this so lution is 9.03. It may be noted that the cost is the output of the objective function in our context.

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	1	0	1	0	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	0	1	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	1	0	0	0	1	2	14
Septic Surgery	0	1	0	0	0	1	7

In addition to evaluating from the cost, we can also evaluate the proposed optimal solution by understanding what fraction of the total requirements have been met. This can be achieved most conveniently by creating a visualization. The following graph represents the fraction of target hours that were satisfied and with the exception of a few departments, most departments have 100% of their requirements met.



We can also evaluate the solution in terms of where the allocated number of rooms lie between the maximum and minimum possible allocations.



4.3 Alternative Solutions

In this section we explore methods to understand if the proposed solution can be further improved by parametrizing the problem differently. To do this, we begin by studying our current solution to see what parameters can be modified. We start by studying the slacks in the current model. The motivation to study the slacks is that some variables may be bound by their upper limits and hence implementing a bound on the cost, hindering it from further reduction. By increasing the upper bounds on the allocations to departments, we may be able to further reduce the cost.

The next table shows the right-hand sides and the slacks in the optimal solution constraints (we do not consider the constraints that have been incorporated in the bounds of the variables). We can see that there are zero slacks in:

Variable	RHS	Slack
0	14	0
1	14	0
2	14	0
3	14	0
4	14	0
5	0	2
6	18	0
7	2	0
8	5	0
9	15	0
10	8	0
11	5	0
12	2	3
13	8	0
14	1	0
15	0	1
16	10	8
17	40	22
18	20	18
19	10	5
20	20	5
21	10	2
22	20	15
23	10	5
24	20	12
25	10	9
26	6	5
27	14	0
28	134.4	8.4
29	67.2	53.2
30	56	21
31	112	7
32	56	0
33	67.2	32.2
34	44.8	9.8
35	89.6	33.6
36	56	49
37	7	0

Table 0.1 Slack

- The total number of operating rooms (constraints *C1*) on Tuesday, Wednesday, and Thursday are at the upper bound of 14 units.
- In the constraint due to the maximum weekly requirements for operating rooms (constraints *C5*) in General Surgery, Orthopedics, Plastic Surgery, and Ophthalmologist, the weekly total is lower.

Therefore, it is interesting to investigate whether an increase in the total number of operating rooms on Tuesday, Wednesday, or Thursday can lead to an improved value of the objective function. We start by increasing the daily total in each day consecutively. The following alternative scenarios are possible.

Scenario 1:

Daily totals constraints: [15, 14, 14, 14, 14]

Optimal solution cost 9.2300 found in 0.0 seconds

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	2	1	1	0	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Scenario 2:

Daily totals constraints: [14, 15, 14, 14, 14]

Optimal solution cost 9.2300 found in 0.0 seconds

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	2	1	1	0	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	0	1	0	0	0	1	7

Scenario 3:

Daily totals constraints: [14, 14, 15, 14, 14]

Optimal solution cost 9.2300 found in 0.0 seconds

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	1	0	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	1	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Scenario 4:

Daily totals constraints: [14, 14, 14, 15, 14]

Optimal solution cost 9.2300 found in 0.0 seconds

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	1	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Scenario 5:

Daily totals constraints: [14, 14, 14, 14, 15]

Optimal solution cost 9.2300 found in 0.0 seconds

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	1	0	1	0	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	0	1	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	1	0	0	2	3	21
Septic Surgery	1	0	0	0	0	1	7

It is observed that there is no improvement by these changes. For example, we can see that in the scenarios, the rooms in a couple of clinics are re-allocated to accommodate the additional room. Still, the total weekly allocation remains the same. The interactions with the other constraints do not allow any improvement. Therefore, seeing no benefit in modifying the parameters, we proceed with the originally proposed optimal solution (which gives the same results).

Going back to the originally proposed solution, we can see that the requested weekly hours' target from each department are:

Weekly Department Requirements	**Weekly Targets**
Pediatric surgery	14
General Surgery	126
Ophthalmologist	50.4
Neurosurgery	49
Thoracic surgery and Cardiac surgery	105
Orthopedics	56
University Surgery	54.6
Otolaryngology	28
Plastic Surgery	56
Urology	35
Septic Surgery	7

Table 0.2 Weekly Department Requirements

The corresponding optimal solution is:

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	1	0	1	0	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	0	1	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	1	0	0	0	1	2	14
Septic Surgery	0	1	0	0	0	1	7

Table 0.3 Optimal Solution

As we can see in the Pediatric Surgery, Orthopedics, and Septic Surgery departments, the requested weekly hours are achieved.

The “optimal solution cost” mentioned in solutions above is the objective function value that the program is trying to maximize. It serves as a metric for evaluation, where higher values are desirable. If the program gives the Weekly Total hours, the “optimal solution cost” is the sum of the percentage of target achievement for each clinic.

In the table below, we can see how the “optimal solution cost” of 9.03 is resulting.

	Weekly Targets (A)	Weekly total (hours) (B)	% Target (B/A)
Pediatric surgery	**14**	**14**	**1.00**
General Surgery	**126**	**126**	**1.00**
Ophthalmologist	**50.4**	**14**	**0.28**
Neurosurgery	**49**	**35**	**0.71**
Thoracic surgery and Cardiac surgery	**105**	**105**	**1.00**
Orthopedics	**56**	**56**	**1.00**
University Surgery	**54.6**	**35**	**0.64**
Otolaryngology	**28**	**28**	**1.00**
Plastic Surgery	**56**	**56**	**1.00**
Urology	**35**	**14**	**0.40**
Septic Surgery	**7**	**7**	**1.00**

Sum	**9.03**

Table 0.4 Optimal Solution Calculation

If the result were 11, then all the departments would reach their targets i.e., a maximum solution cost that is attainable is 11.

If a department or the hospital changes the input data, the program will run a new scenario.

The program above along with data files can be found and executed as a binder notebook by clicking here.

Conclusion and Remarks

The paper discussed in detail how the Papageorgiou hospital in Greece can optimally allocate operating rooms to departments to maximize the fulfilled requirements. 55 decision variables were tweaked to maximize the allocation. Once an initial optimal solution was achieved, we attempted to improve the fraction of requirements fulfilled by changing the upper and lower bounds on the decision variables (found in the constraints). However, doing so did not give results any better than the initial optimization which may serve to explain that our optimal solution is not unique, but the global maximum value of the objective function may have been achieved, nonetheless.

Having solved the integer optimization problem, we can go one step forward by giving each clinic an average duration for one surgery, then with the available hours we took from the program, we can see how many surgeries the department will have daily and weekly. Summing all these, the hospital can have a forecast as well.

Department	Av. Time per surgery (min)	Weekly total (hours)	Av. Surgeries per week		Av Surgeries Per day
Pediatric surgery	60	14	14		3
General Surgery	75	126	101		20
Ophthalmologist	60	14	14		3
Neurosurgery	120	35	18		4
Thoracic surgery and Cardiac surgery	150	105	42		8
Orthopedics	60	56	56		11
University Surgery	60	35	35		7
Otolaryngology	75	28	22		4
Plastic Surgery	60	56	56		11
Urology	60	14	14		3
Septic Surgery	45	7	9		2
	Average weekly Surgeries:		381	Average daily Surgeries:	76

Table 0.1 Average Surgeries per Clinic

Depending on the kind and the duration of each surgery for the clinics, the daily and consequently the weekly number of surgeries is adjusted. Approximately, there is the possibility of about 20,000 surgeries per year.

The benefit of the operating room scheduling is that the flow of the surgeries will be smooth, not only for the hospital but between the clinics. Improved operating room scheduling could reduce costs by reducing underutilization and/or reducing the workday's unplanned extension (unexpected overtimes). The benefits of better scheduling would also extend to patients by organizing appointments better and on time, informing them earlier, thus reducing unpredictable and unnecessary waiting times.

Appendix 1.1

SOLVING WITH PYTHON

This appendix section dives into the details of using python to solve the problem step by step using python.

We collect and insert the data of our problem in an Excel file creating tables. From this Excel file, the program will read and calculate. One table is Table 4.6.1 Number of Surgical Teams per Department per Day that we mention before in Chapter 3 and the others are Table 4.6.2 Minimum number of ORs per day, Table 4.6.3 Maximum number of ORs per day, the Table 4.6.4 Requirement's compare tables and the Table 4.6.5 Weekly Targets per Department.

We prepare the program

```
## Uncomment the lines below to install the necessary libraries in
 binder

#import sys
#!{sys.executable} -m pip install pandas
#!{sys.executable} -m pip install matplotlib
#!{sys.executable} -m pip install scipy
#!{sys.executable} -m pip install mip
#!{sys.executable} -m pip install xlrd


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import font_manager as fm, rcParams

rcParams["font.size"] = "12"


from matplotlib import colors as mcolors
colors = dict(mcolors.BASE_COLORS, **mcolors.CSS4_COLORS)

import os
import re
import string
from mip import *
import time

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

First, we import the data from the Excel file into a Pandas dataframe data. For convenience, we rename the columns as in the Excel file, using upper-case letters.

```
data = pd.read_excel('ORscheduling_Complete (1).xlsx',header=None)

new_names = string.ascii_uppercase[:len(data.columns)] ## as many
letters as needed
rename_dict = dict(zip(data.columns,new_names))

data.rename(columns=rename_dict,inplace=True)
```

Part of the Pandas dataframe *data* is shown below.

```
data.head(32)
data.tail(1)
#data
```

5	NaN	M	T	W	R	F
6	Pediatric surgery	1	2	1	2	1
7	General Surgery	4	4	8	4	4
8	Ophthalmologist	2	2	2	4	2
9	Neurosurgery	2	2	2	2	2
10	Thoracic surgery and Cardiac surgery	4	4	4	4	4
11	Orthopedics	2	2	2	2	2
12	University Surgery	2	2	2	2	4
13	Otolaryngology	2	1	2	1	2
14	Plastic Surgery	4	2	4	2	4
15	Urology	2	2	2	2	2
16	Septic Surgery	1	1	1	1	2
17	NaN	26	24	30	26	29

Next, we import the various regions containing the parameters of the problem into Numpy arrays. Note that, contrary to the usual convention in Python, the range or rows defined in each call to *data.loc()*, say *data.loc(n :n ,'A')* includes the last row defined in the range, i.e. n_2 this case.

- The **departments**:

```
print('Departments:\n')
departments = data.loc[6:16,'A'].values
print(departments)
```

The result we get:

```
Departments:

['Pediatric surgery' 'General Surgery' 'Ophthalmologist' 'Neur
osurgery'
 'Thoracic surgery and Cardiac surgery' 'Orthopedics' 'Univers
ity Surgery'
 'Otolaryngology' 'Plastic Surgery' 'Urology' 'Septic Surgery'
]
```

- The **available surgical teams**, per department, per day:

```
## Surgical Teams per Department per Day
print('\nTeams availability per department (rows) per day (col
umns):\n')
available_teams = data.loc[6:16,'B':'F'].values.astype(int)
print(available_teams)
```

And we get from the program:

```
Teams availability per department (rows) per day (columns):

[[1 2 1 2 1]
 [4 4 8 4 4]
 [2 2 2 4 2]
 [2 2 2 2 2]
 [4 4 4 4 4]
 [2 2 2 2 2]
 [2 2 2 2 4]
 [2 1 2 1 2]
 [4 2 4 2 4]
 [2 2 2 2 2]
 [1 1 1 1 2]]
```

Exactly like the table we insert to Excel file. Then, we continue with:

- The **minimum and maximum numbers of operational rooms**, per department, per day:

```
## Min # of ORs per day
print('\nMinimum number of ORs per department (rows) per day (
columns):\n')
min_ors = data.loc[6:16,'H':'L'].values.astype(int)
print(min_ors)

## Max # of ORs per day
print('\nMaximum number of ORs per department (rows) per day (
columns):\n')
max_ors = data.loc[6:16,'N':'R'].values.astype(int)
print(max_ors)
```

And the out of the program:

```
Minimum number of ORs per department (rows) per day (columns):

[[0 0 0 0 0]
 [3 3 6 3 3]
 [0 0 0 2 0]
 [1 1 1 1 1]
 [3 3 3 3 3]
 [2 2 1 2 1]
 [1 1 0 1 2]
 [1 0 1 0 0]
 [2 1 2 1 2]
 [0 0 0 0 1]
 [0 0 0 0 0]]

Maximum number of ORs per department (rows) per day (columns):

[[2 2 2 2 2]
 [8 8 8 8 8]
 [4 4 4 4 4]
 [2 2 2 2 2]
 [4 4 4 4 4]
 [2 2 2 2 2]
 [4 4 4 4 4]
 [2 2 2 2 2]
 [4 4 4 4 4]
 [2 2 2 2 2]
 [1 1 1 1 2]]
```

- The **minimum and maximum weekly department requirements for operational rooms**, per department:

```
## Weekly Department Requirements
print('\nWeekly department requirements (1st column: min/ 2nd
column: max):\n')
```

```
weekly_reqs = data.loc[21:31,'B':'C'].values.astype(int)
print(weekly_reqs)
```

```
Weekly department requirements (1st column: min/ 2nd column:
max):

[[ 0 10]
 [18 40]
 [ 2 20]
 [ 5 10]
 [15 20]
 [ 8 10]
 [ 5 20]
 [ 2 10]
 [ 8 20]
 [ 1 10]
 [ 0  6]]
```

- And finally, the **weekly targets per department**, in hours per week:

```
## Weekly Department Targets
print('\nWeekly targets per department (hours):\n')
weekly_targets = data.loc[21:31,'H'].values.astype(float)
print(weekly_targets)
```

```
Weekly targets per department (hours):

[ 14.  126.   50.4  49.  105.   56.   54.6  28.   56.   35.
  7. ]
```

Appendix 1.2

MODELING THE PROBLEM USING THE MIP PACKAGE COMMANDS

We use the Python mip package (https://docs.python-mip.com/en/latest/intro.html). First, we initialize a model *m*, we define the direction of the optimization (*MAXIMIZE*) and select the solver (the default, *CBC*). *CBC* stands for Coin-or branch and cut, and is an open-source Mixed Integer Programming (MIP) library using the branch-and-cut approach (https://en.wikipedia.org/wiki/Branch_and_cut), i.e. combining the classic branch – and – bound method with cutting planes for the tightening of the bounds.

Next, we define the 55 decision variables and give them names x_{i_j} using the function *add_var()*. In the same command, we select their type as *INTEGER* and also set their lower and upper bounds – from constraints *(C_3)* and *(C_4)* - using the arrays *min_ors* and *max_ors*. We also incorporate constraints *(C_2)* (operating rooms are ≤ available surgical teams), in the upper bounds, using array *available_teams* and use the *minimum* (i.e. tighter) of the two upper bounds.

Note that we use subscripts for the departments (i) and days (j) from 1 to 5. Therefore, when using the other data of the problem, we deduct one from i and/or j.

Finally, we define the objective function (property *objective*), according to the formulation in section **B**, using data from array *weekly_targets*. Note that we use the function *var_by_name()* to refer to a variable by its name rather than by an index. Also, we use the function *xsum()* to add several variables together.

```
m = Model(name='Operation Rooms',sense=MAXIMIZE, solver_name=C
BC)


## Daily # ORs assigned is integer
x = [m.add_var(name='x'+str(i)+'_'+str(j),var_type=INTEGER, \
               lb = min_ors[i-1,j-1], ub= min(max_ors[i-1,j-1]
,available_teams[i-1,j-1])) \
                 for i in range(1,12) for j in range(1,6)]

m.objective = xsum(xsum(7/weekly_targets[i-1]*m.var_by_name('x
'+str(i)+'_'+str(j)) \
                 for j in range(1,6)) for i in range(1,12))
```

Next we add the first constraints (C_1) (total number of operating rooms each day ≤ 14). We use the += shorthand notation for the function *add_constr()* and again use function *xsum()* to add several variables together.

As with the objective function, we refer to variables using their names, with the function *var_by_name()*.

```
## Total # ORs each day
## the total number of operating rooms each day has to be less
 than or equal to 14.
for j in range(1,6):
    m += xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for i in ra
nge(1,12)) <= 14
```

The next constraints *(C_5)* and *(C_6)* are very similar and express the constraints due to the minimum / maximum weekly requirements for operating rooms. We use array *weekly_reqs* where the first column (0) has the minimum values and the second column (1) has the maximum ones.

Finally, we add constraints *(C_7)* which put upper bounds (weekly targets) on the weekly numbers of operating rooms, for each department. The values for the right side are taken from array *weekly_targets*.

```
## UB: Weekly Dept OR requirement
## LB: Weekly Dept OR requirement
for i in range(1,12):
    m += xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for j in
range(1,6)) >= weekly_reqs[i-1,0] #,'week_'+str(j)+' min'
for i in range(1,12):
    m += xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for j in
range(1,6)) <= weekly_reqs[i-1,1] #,'week_'+str(j)+' max'


## -----------------------------------------------------------
-------
## Departmental Targets
for i in range(1,12):
    m += 7*xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for j i
n range(1,6)) <= weekly_targets[i-1] #,'weekly_target_'+str(
i)
```

We check the model by writing it in *lp format* in a file *model.lp*. We read back this file and examine it (see next output). We could have written the model also in the more portable *mps format* (by only changing the file extension to *.mps*) but this latter format is more difficult to read.

We can see that, internally, the problem is turned into a minimization one, by changing the sign of the coefficients of the objective function.

```
m.write('model.lp')
with open('model.lp', 'r') as f:
    model = f.readlines()
for line in model:
    print(re.sub('\n','',line))
```

```
\Problem name: Operation Rooms

Minimize
```

```
OBJROW: -0.50000 x1_1 -0.50000 x1_2 -0.50000 x1_3 -0.50000 x1_
4 -0.50000 x1_5 -0.05556 x2_1 -0.05556 x2_2 -0.05556 x2_3 -0.0
5556 x2_4 -0.05556 x2_5
 -0.13889 x3_1 -0.13889 x3_2 -0.13889 x3_3 -0.13889 x3_4 -0.13
889 x3_5 -0.14286 x4_1 -0.14286 x4_2 -0.14286 x4_3 -0.14286 x4
_4 -0.14286 x4_5
 -0.06667 x5_1 -0.06667 x5_2 -0.06667 x5_3 -0.06667 x5_4 -0.06
667 x5_5 -0.12500 x6_1 -0.12500 x6_2 -0.12500 x6_3 -0.12500 x6
_4 -0.12500 x6_5
 -0.12821 x7_1 -0.12821 x7_2 -0.12821 x7_3 -0.12821 x7_4 -0.12
821 x7_5 -0.25000 x8_1 -0.25000 x8_2 -0.25000 x8_3 -0.25000 x8
_4 -0.25000 x8_5
 -0.12500 x9_1 -0.12500 x9_2 -0.12500 x9_3 -0.12500 x9_4 -0.12
500 x9_5 -0.20000 x10_1 -0.20000 x10_2 -0.20000 x10_3 -0.20000
 x10_4 -0.20000 x10_5
 - x11_1 - x11_2 - x11_3 - x11_4 - x11_5
Subject To
constr(0):  x1_1 + x2_1 + x3_1 + x4_1 + x5_1 + x6_1 + x7_1 + x
8_1 + x9_1 + x10_1
 + x11_1 <= 14
constr(1):  x1_2 + x2_2 + x3_2 + x4_2 + x5_2 + x6_2 + x7_2 + x
8_2 + x9_2 + x10_2
 + x11_2 <= 14
constr(2):  x1_3 + x2_3 + x3_3 + x4_3 + x5_3 + x6_3 + x7_3 + x
8_3 + x9_3 + x10_3
 + x11_3 <= 14
constr(3):  x1_4 + x2_4 + x3_4 + x4_4 + x5_4 + x6_4 + x7_4 + x
8_4 + x9_4 + x10_4
 + x11_4 <= 14
constr(4):  x1_5 + x2_5 + x3_5 + x4_5 + x5_5 + x6_5 + x7_5 + x
8_5 + x9_5 + x10_5
 + x11_5 <= 14
constr(5):  x1_1 + x1_2 + x1_3 + x1_4 + x1_5 >= -0
constr(6):  x2_1 + x2_2 + x2_3 + x2_4 + x2_5 >= 18
constr(7):  x3_1 + x3_2 + x3_3 + x3_4 + x3_5 >= 2
constr(8):  x4_1 + x4_2 + x4_3 + x4_4 + x4_5 >= 5
constr(9):  x5_1 + x5_2 + x5_3 + x5_4 + x5_5 >= 15
constr(10):  x6_1 + x6_2 + x6_3 + x6_4 + x6_5 >= 8
constr(11):  x7_1 + x7_2 + x7_3 + x7_4 + x7_5 >= 5
constr(12):  x8_1 + x8_2 + x8_3 + x8_4 + x8_5 >= 2
constr(13):  x9_1 + x9_2 + x9_3 + x9_4 + x9_5 >= 8
constr(14):  x10_1 + x10_2 + x10_3 + x10_4 + x10_5 >= 1
constr(15):  x11_1 + x11_2 + x11_3 + x11_4 + x11_5 >= -0
constr(16):  x1_1 + x1_2 + x1_3 + x1_4 + x1_5 <= 10
constr(17):  x2_1 + x2_2 + x2_3 + x2_4 + x2_5 <= 40
constr(18):  x3_1 + x3_2 + x3_3 + x3_4 + x3_5 <= 20
constr(19):  x4_1 + x4_2 + x4_3 + x4_4 + x4_5 <= 10
constr(20):  x5_1 + x5_2 + x5_3 + x5_4 + x5_5 <= 20
constr(21):  x6_1 + x6_2 + x6_3 + x6_4 + x6_5 <= 10
constr(22):  x7_1 + x7_2 + x7_3 + x7_4 + x7_5 <= 20
constr(23):  x8_1 + x8_2 + x8_3 + x8_4 + x8_5 <= 10
constr(24):  x9_1 + x9_2 + x9_3 + x9_4 + x9_5 <= 20
constr(25):  x10_1 + x10_2 + x10_3 + x10_4 + x10_5 <= 10
constr(26):  x11_1 + x11_2 + x11_3 + x11_4 + x11_5 <= 6
constr(27):  7 x1_1 + 7 x1_2 + 7 x1_3 + 7 x1_4 + 7 x1_5 <= 14
constr(28):  7 x2_1 + 7 x2_2 + 7 x2_3 + 7 x2_4 + 7 x2_5 <= 126
```

```
constr(29):  7 x3_1 + 7 x3_2 + 7 x3_3 + 7 x3_4 + 7 x3_5 <= 50.
40000
constr(30):  7 x4_1 + 7 x4_2 + 7 x4_3 + 7 x4_4 + 7 x4_5 <= 49
constr(31):  7 x5_1 + 7 x5_2 + 7 x5_3 + 7 x5_4 + 7 x5_5 <= 105
constr(32):  7 x6_1 + 7 x6_2 + 7 x6_3 + 7 x6_4 + 7 x6_5 <= 56
constr(33):  7 x7_1 + 7 x7_2 + 7 x7_3 + 7 x7_4 + 7 x7_5 <= 54.
60000
constr(34):  7 x8_1 + 7 x8_2 + 7 x8_3 + 7 x8_4 + 7 x8_5 <= 28
constr(35):  7 x9_1 + 7 x9_2 + 7 x9_3 + 7 x9_4 + 7 x9_5 <= 56
constr(36):  7 x10_1 + 7 x10_2 + 7 x10_3 + 7 x10_4 + 7 x10_5 <
= 35
constr(37):  7 x11_1 + 7 x11_2 + 7 x11_3 + 7 x11_4 + 7 x11_5 <
= 7
Bounds
 0 <= x1_1 <= 1
 0 <= x1_2 <= 2
 0 <= x1_3 <= 1
 0 <= x1_4 <= 2
 0 <= x1_5 <= 1
 3 <= x2_1 <= 4
 3 <= x2_2 <= 4
 6 <= x2_3 <= 8
 3 <= x2_4 <= 4
 3 <= x2_5 <= 4
 0 <= x3_1 <= 2
 0 <= x3_2 <= 2
 0 <= x3_3 <= 2
 2 <= x3_4 <= 4
 0 <= x3_5 <= 2
 1 <= x4_1 <= 2
 1 <= x4_2 <= 2
 1 <= x4_3 <= 2
 1 <= x4_4 <= 2
 1 <= x4_5 <= 2
 3 <= x5_1 <= 4
 3 <= x5_2 <= 4
 3 <= x5_3 <= 4
 3 <= x5_4 <= 4
 3 <= x5_5 <= 4
 2 <= x6_1 <= 2
 2 <= x6_2 <= 2
 1 <= x6_3 <= 2
 2 <= x6_4 <= 2
 1 <= x6_5 <= 2
 1 <= x7_1 <= 2
 1 <= x7_2 <= 2
 0 <= x7_3 <= 2
 1 <= x7_4 <= 2
 2 <= x7_5 <= 4
 1 <= x8_1 <= 2
 0 <= x8_2 <= 1
 1 <= x8_3 <= 2
 0 <= x8_4 <= 1
 0 <= x8_5 <= 2
 2 <= x9_1 <= 4
 1 <= x9_2 <= 2
 2 <= x9_3 <= 4
```

```
 1 <= x9_4 <= 2
 2 <= x9_5 <= 4
 0 <= x10_1 <= 2
 0 <= x10_2 <= 2
 0 <= x10_3 <= 2
 0 <= x10_4 <= 2
 1 <= x10_5 <= 2
 0 <= x11_1 <= 1
 0 <= x11_2 <= 1
 0 <= x11_3 <= 1
 0 <= x11_4 <= 1
 0 <= x11_5 <= 2
Integers
x1_1 x1_2 x1_3 x1_4 x1_5 x2_1 x2_2 x2_3 x2_4 x2_5
x3_1 x3_2 x3_3 x3_4 x3_5 x4_1 x4_2 x4_3 x4_4 x4_5
x5_1 x5_2 x5_3 x5_4 x5_5 x6_1 x6_2 x6_3 x6_4 x6_5
x7_1 x7_2 x7_3 x7_4 x7_5 x8_1 x8_2 x8_3 x8_4 x8_5
x9_1 x9_2 x9_3 x9_4 x9_5 x10_1 x10_2 x10_3 x10_4 x10_5
x11_1 x11_2 x11_3 x11_4 x11_5
End
```

In the following code we wrap the entire modeling in a function *make_model()* with arguments the parameters of the problem. We do this because we intend to change some coefficients in the end, and it is not possible in *mip* to make such changes directly in a model.

In this function, we replace the constant in constraints *(C_1)* (maximum total number of operating rooms each day) by a vector *daily_totals_upper*.

```
def make_model(min_ors,max_ors,available_teams,weekly_targets, d
aily_totals_upper,weekly_reqs):
    m = Model(name='Operation Rooms',sense=MAXIMIZE, solver_name
=CBC)
    ## Daily # ORs assigned is integer
    x = [m.add_var(name='x'+str(i)+'_'+str(j),var_type=INTEGER,
\
                   lb = min_ors[i-1,j-1], ub= min(max_ors[i-1,j-
1],available_teams[i-1,j-1])) \
                   for i in range(1,12) for j in range(1,6)]

    m.objective = xsum(xsum(7/weekly_targets[i-1]*m.var_by_name(
'x'+str(i)+'_'+str(j)) \
                             for j in range(1,6)) for i in range(
1,12))
    ## Total # ORs each day
    ## the total number of operating rooms each day has to be le
ss than or equal to 14.
    for j in range(1,6):
        m += xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for i in
range(1,12)) <= daily_totals_upper[j-1]
```

```
    ## UB: Weekly Dept OR requirement
    ## LB: Weekly Dept OR requirement
    for i in range(1,12):
        m += xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for j in
range(1,6)) >= weekly_reqs[i-1,0] #,'week_'+str(j)+' min'
    for i in range(1,12):
        m += xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for j in
range(1,6)) <= weekly_reqs[i-1,1] #,'week_'+str(j)+' max'


    ## ----------------------------------------------------------
--------
    ## Departmental Targets
    for i in range(1,12):
        m += 7*xsum(m.var_by_name('x'+str(i)+'_'+str(j)) for j i
n range(1,6)) <= weekly_targets[i-1] #,'weekly_target_'+str(i)

    return m
```

Next, we select some parameters for the search:

- SearchEmphasis = 1 (FEASIBILITY) selects a strategy which puts priority to finding quickly feasible solutions and improve them, rather than going directly for an optimal solution.
- max_gap = 0.05 sets the allowable percentage deviation of the upper bound from the lower bound for concluding the search.
- preprocess = 1 enables a procedure which tries to improve the formulation.

Finally, we initiate the search process with the function *optimize()*, assigning the result of calling this function to a variable *status*.

```
SearchEmphasis = 1 ; LP_Method = 1
m.max_gap = 0.05
m.preprocess = 1

start_time = time.time()

status = m.optimize()
```

The commands below report the outcome of the search, depending on the value of the variable *status*. The value in our case is equal to the constant *OptimizationStatus.OPTIMAL*, meaning that the solution is optimal.

We can then report the value of property *objective_value* (the objective function value) and the values of the decision variables at the optimal solution (properties *name* and *x* for each variable in the list *vars*).

```
if status == OptimizationStatus.OPTIMAL:
    print('optimal solution cost {0:6.2f} found in {1:8.1f} se
conds'.\
          format(round(m.objective_value,2),(time.time() - sta
rt_time)))

elif status == OptimizationStatus.FEASIBLE:
    print('sol.cost {} found, best possible: {}'.format(m.obje
ctive_value, m.objective_bound))
elif status == OptimizationStatus.NO_SOLUTION_FOUND:
    print('no feasible solution found, lower bound is: {}'.for
mat(m.objective_bound))
elif status == OptimizationStatus.INFEASIBLE:
    print('no feasible solution found')
if status == OptimizationStatus.OPTIMAL or status == Optimizat
ionStatus.FEASIBLE:
    print('solution:')
    for v in m.vars:
        print('{0:5s}: {1:3d}'.format(v.name, int(v.x)))
```

Appendix 1.3

IMPROVING THE CURRENT SOLUTION ON PYTHON USING MAKE_MODEL()

To check if increasing the upper bounds can lead to improvement, we start by increasing the daily total in each day consecutively. We use the function *make_model()* created before, clearing the model each time, by calling the function *clear()*.

```
initial_daily = [14]*5
for j in range(5):
    new_daily = initial_daily.copy()
    new_daily[j] += 1
    print('\n--------------------\nDaily totals constraints: '
,new_daily,'\n')

    m.clear() ; m = make_model(min_ors,max_ors,available_teams
,weekly_targets,new_daily,weekly_reqs)
    SearchEmphasis = 1 ; LP_Method = 1
    m.max_gap = 0.05
```

```
    m.preprocess = 1

    start_time = time.time()
    status = m.optimize()
    if status == OptimizationStatus.OPTIMAL:
        print('optimal solution cost {0:8.4f} found in {1:8.1f
} seconds'.\
              format(round(m.objective_value,2),(time.time()
- start_time)))
        results = pd.DataFrame([[m.var_by_name('x'+str(i)+'_'+
str(j)).x for j in range(1,6)] for i in range(1,12)],
                        index = departments, columns = ['Mon','
Tue','Wed','Thu','Fri'],dtype=int)
        results['Weekly total'] = results.apply(sum, axis=1)
        results['Weekly total (hours)'] = 7*results['Weekly to
tal']
        results
        print('Daily totals:\n')
        print(results.apply(sum, axis=0)[:-2].values)

    else:
        print('Failed.')
```

```
--------------------
Daily totals constraints:  [15, 14, 14, 14, 14]

optimal solution cost   9.2300 found in      0.0 seconds
```

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	2	1	1	0	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Table 0.1 Alternative Scenario 1

```
Daily totals:

[15 14 14 14 14]

--------------------
Daily totals constraints:  [14, 15, 14, 14, 14]

optimal solution cost   9.2300 found in      0.0 seconds
```

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	2	1	1	0	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	0	1	0	0	0	1	7

Table 0.3 Alternative Scenario 2

```
Daily totals:

[14 15 14 14 14]

--------------------
Daily totals constraints:  [14, 14, 15, 14, 14]

optimal solution cost   9.2300 found in      0.0 seconds
```

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	1	0	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	1	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Table 0.2 Alternative Scenario 3

```
Daily totals:

[14 14 15 14 14]

--------------------
Daily totals constraints:  [14, 14, 14, 15, 14]

optimal solution cost   9.2300 found in      0.0 seconds
```

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	1	0	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Table 0.3 Alternative Scenario 4

```
Daily totals:

[14 14 14 15 14]

--------------------
Daily totals constraints:  [14, 14, 14, 14, 15]

optimal solution cost   9.2300 found in      0.0 seconds
```

	Mon	Tue	Wed	Thu	Fri	Weekly total	Weekly total (hours)
Pediatric surgery	0	0	0	1	1	2	14
General Surgery	3	3	6	3	3	18	126
Ophthalmologist	0	0	0	2	0	2	14
Neurosurgery	1	1	1	1	1	5	35
Thoracic surgery and Cardiac surgery	3	3	3	3	3	15	105
Orthopedics	2	2	1	2	1	8	56
University Surgery	1	1	0	1	2	5	35
Otolaryngology	1	1	1	0	1	4	28
Plastic Surgery	2	1	2	1	2	8	56
Urology	0	2	0	0	1	3	21
Septic Surgery	1	0	0	0	0	1	7

Table 0.4 Alternative Scenario 5

```
Daily totals:

[14 14 14 14 15]
```

REFERENCES

A., M., 2006. Are Your Hospital Operating Rooms "Efficient"?.

B. Cardoen, E. D. a. J. B., 2006. Optimizing a multiple objective surgical case scheduling problem.. *Working Paper*.

Bertsimas, D., 2017. *The Analytics Edge,* s.l.: Massachusetts Institute of Technology: MIT OpenCourseWare.

Bishop, W. J., 1960. The early history of Surgery.

Dexter., F., 2000. A strategy to decide whether to move the last case of the day in an operating. *Anesthesia and Analgesia*.

Gawande, A., 2012. Two Hundred Years of Surgery. *The New England Journal of Medicine*, 3 May.

Hillier, F. S. a. G. J. L. J., 2004. "Introduction to Operations Research.".

Hockstein, N. G., 2007. "A history of robots: from science fiction to surgical robotics.". *Journal of robotic surgery* .

MA., F., 2015. Neolithic trepanation decoded- A unifying hypothesis: Has the mystery as to why primitive surgeons performed cranial surgery been solved?.

Marcon, E. P. & Dexter, F. M. P., 2007. An Observational Study of Surgeons' Sequencing of Cases and Its Impact on Postanesthesia Care Unit and Holding Area Staffing Requirements at Hospitals. *Anesthesia & Analgesia*, July.

P., P., 1995. A History of Medicine. Primitive and Ancient Medicine..

Peter VanBerkel, J. T. B., 2012. Quantitative Modelling for Wait Time Reduction.

Ribeiro, J. F. L. A. &. C. A., 2018. A two-phase method for operating room scheduling.. *Brazilian Journal of Operations & Production Management*.