ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF INDUSTRIAL ENGINEERING

# Project Report

## 2021-2022 Spring

# *Scheduling for a Local Hospital*

**Taylan Bektaş**

**Berkant Daş**

# Table of Contents

# 1 Introduction

Scheduling is one of the key elements in almost every industry. In this study, a scheduling problem is examined for METU Hospital. The hospital aims to schedule medical operations for daily surgery rooms. Schedules are done on five consecutive days. Surgeries that are performed each day are determined and scheduled according to some constraints. An effective MATLAB scheduling algorithm aims to construct a method to maximize the number of patients and utilize the surgery rooms. There are 2 schedule objectives in this study. The first objective is to maximize the number of operations which is completed within its initial available time interval. The second objective is to maximize the utilization of the rooms. In the rest of the report, these two schedule objectives will be examined in a detailed way.

# 2 Constraints

- There is a certain number of surgery rooms in the hospital which is 3 and 4 for the cases in the study, and the operations are placed only in one of these rooms. Only one operation will be performed at a given time in each room. In other words, operations in the same room cannot overlap. There are 97 patients.

- The time horizon for scheduling is between 09:00 and 17:00 for each day. The interval has been taken as [0, 480].

- Each operation has a fixed time and needs to be completed within the given timeframe.

- Every patient has a priority level, represented by a 4-level integer value as 1, 2, 3, and 4. Level 1 denotes the highest priority, and Level- 4 indicates the least priority. The patient with the higher priority should spend less waiting time in the system than those with lower priorities.

- When an operation cannot be scheduled on its original day, it is scheduled the next day by changing the priority level to 0, which indicates an emergency for the patient.

# 3 Assumptions

- The Gantt charts show three color codes shown *blue*, *yellow*, *green,* and *white*.

- The color *blue* represents a surgery operation on the interval it fulfills.

- The color *yellow* represents an Available Interval for the operations of the patient written on the interval. The operation can take place anywhere on the *yellow* interval but occur in a specific place due to the algorithm's aims.

- The color *green* represents an Available Interval for the operations of both patients written on the interval. Rather than placing two *yellow* layers on top of each other, *green* is used for ease of understanding. The operation can take place anywhere on the *green* interval but occur in a specific place due to the algorithm's aims.

- For any patient, when the patient's operation takes any place on his/her or another patient's available interval, the color *blue,* which indicates the operation, overwrites the color *yellow* and *green*, whose meanings are explained above.

- If there is no operation or available interval on that specific interval, it is left *white*. There can not be any operation performed on the intervals that have the color *white*.

- Only in Gantt charts' titles, it is assumed that the maximum placement is Objective 2 and the maximum utility is Objective 1, in contrast to the rest of the report.

- To support the provided claims, sometimes utility matrixes are used throughout the report. In these matrices, while the row numbers represent the day numbers, column numbers represent the room numbers.

- The increasing room number by one condition is satisfied as explained. In the main script, in the beginning, it is asked whether the room number is increased by one or not by inputting 1. If input 0, then the room number is going to be 3; otherwise, 4.

# 4 Analyzing the Algorithm

The two schedule objectives of the algorithm are explained in the introduction part. In this section, both the objectives will be examined in terms of objective function value, utilization of the rooms, and the number of operations shifted in each priority level.

In the first algorithm, which provides maximum placement, the general algorithm is as below. Operations are getting in order, from 1st priority to 4th priority, and from higher duration to smaller duration. Then operations are scheduled in the same order. The algorithm determines the ways of scheduling. For instance, first, it looks at whether the operation can be scheduled adjacent to the other scheduled operations. If this cannot be found, then it goes to the second step. In the second step, the algorithm looks at whether the available interval is closer to minute 0 or minute 480. Then, for instance, if the available interval is closer to 0, the operation is placed to the leftmost of the available interval. Same for the other side. Of course, the suitability of the leftmost and rightmost intervals are checked.

In the second algorithm which provides utility maximization, the algorithm repeats the same ways as the maximum placement algorithm until the second step. After the second step, the algorithm automatically places the operation on the rightmost available interval while checking its suitability.

## 4.1 Schedule Objective 1 – Maximum Placement

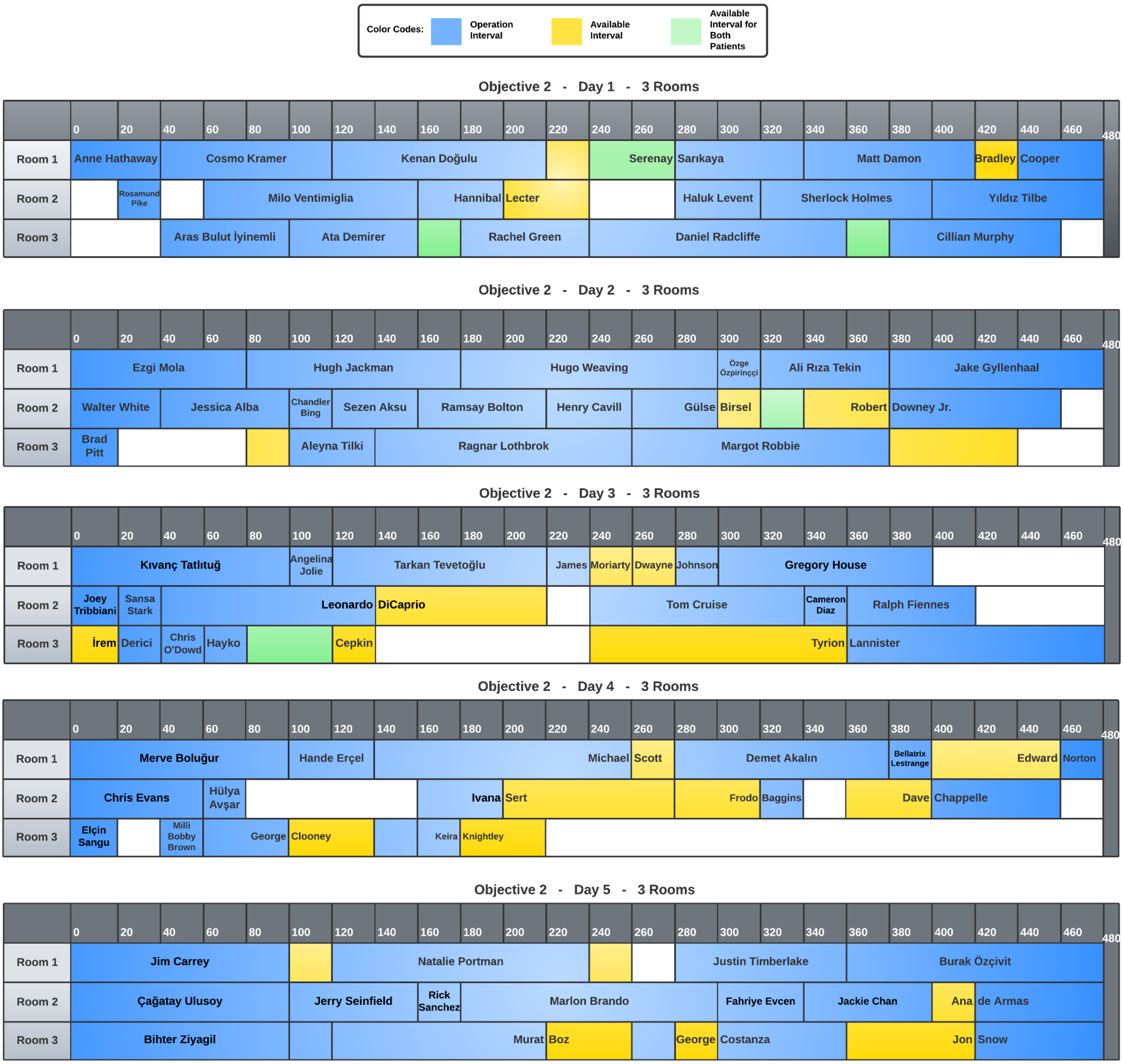
The schedule objective function value is calculated by subtracting the number of patients who couldn't have the surgery from the number of patients.

$$Function\ Value = Number\ of\ Patients - Number\ of\ Patients\ Couldn't\ Get\ Operated$$

In this case, the number of patients is 97 and the number of patients who couldn't get operated due to postponing is 16. Therefore, the schedule objective function value is 81.

Unlike the maximum utility algorithm, the maximum placement algorithm aims to place the maximum number of patients rather than focusing primarily on room utilization. Below is a

Gantt chart for the maximum placement algorithm when the number of rooms is 3 with five consecutive days.




Gantt Chart of Maximum Placement Algorithm with 3 Rooms

As seen in Figure 1, this time, the utility of the rooms is not as high as the maximum utility algorithm. Instead, it focuses on increasing the number of patients placed.

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.8333 | 0.7500 | 0.7917 |
| 2 | 1 | 0.7917 | 0.6250 |
| 3 | 0.7500 | 0.6667 | 0.3750 |
| 4 | 0.8333 | 0.4167 | 0.2083 |
| 5 | 0.8750 | 0.9583 | 0.6667 |

Above is the matrix that shows the utilities of the room corresponding to the days. Columns are rooms.

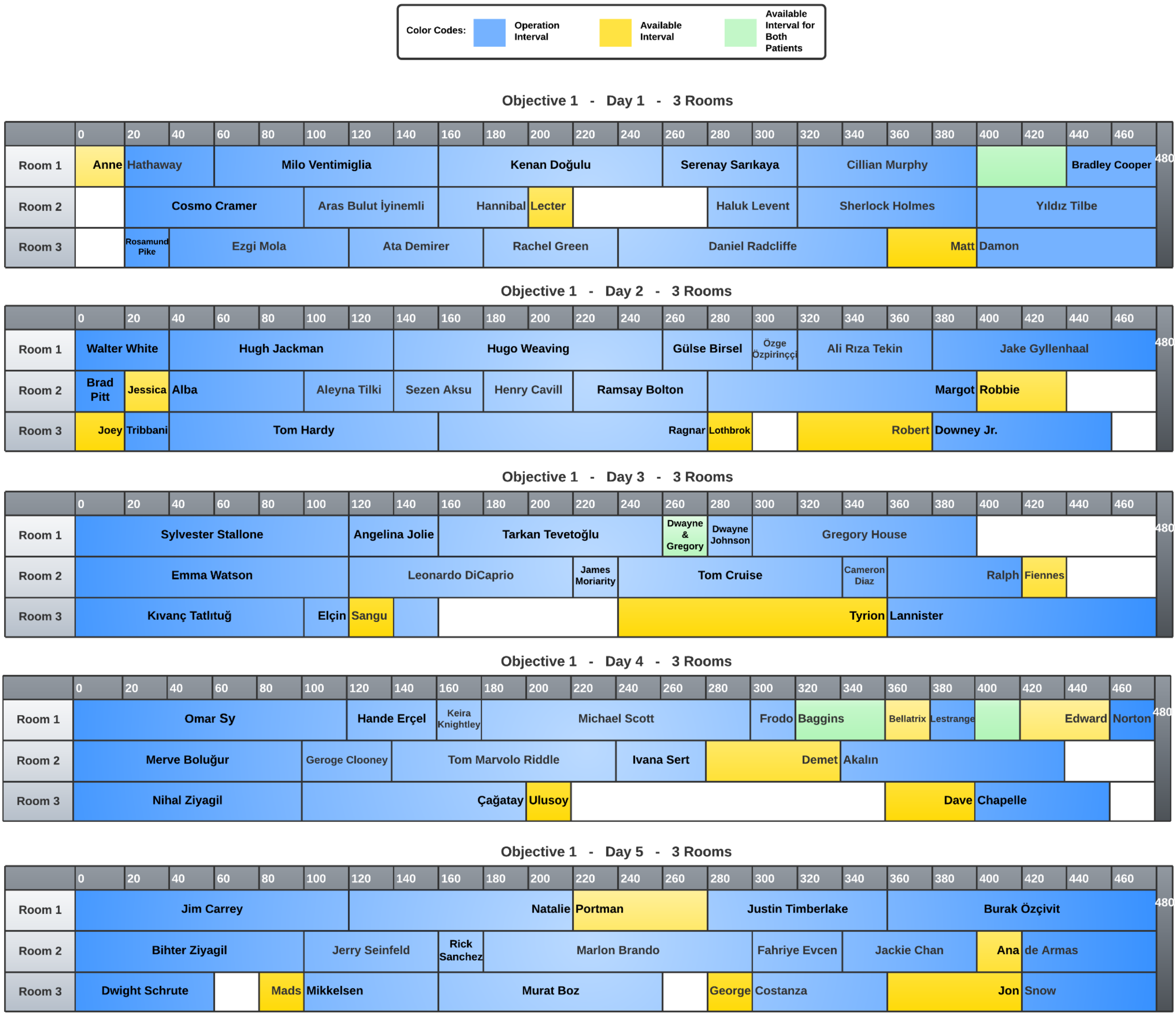
### 4.2 Schedule Objective 2 – Maximum Utility

As explained in the above subsection 4.1, the schedule objective function value can be calculated to analyze the algorithm. In the maximum placement algorithm, the number of patients who could not get operated on due to postponing is 16, and the number of patients is 97. Therefore, the schedule objective functiın value is 81.

Since the aim of this algorithm is to utilize the usage of the rooms, examining the utilization is crucial. In this study, what is meant by utilization is the usage of the rooms in a most efficient way. In other words, rooms are utilized if the rooms are used in a way that maximum usage of some rooms enables others to be available. Below is a Gantt chart of the algorithm output of the maximum utility algorithm when the number of rooms is 3 with five consecutive days.

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.8750 | 0.7917 | 0.8750 |
| 2 | 1 | 0.7917 | 0.7083 |
| 3 | 0.7500 | 0.8750 | 0.5000 |
| 4 | 0.7500 | 0.7917 | 0.5417 |
| 5 | 0.8750 | 0.9583 | 0.7083 |

Above is the matrix that shows the utilities of the room corresponding to the days. Columns are rooms.




Gantt Chart of Maximum Utility Algorithm with 3 Rooms

As clearly seen in *Figure 2*, the Gantt chart is an optimal instrument for visualizing the output data. *Figure 2* shows that the full placement of some rooms enables other rooms to be used less, which can be observed by the usage frequency of the color *white*.

# 5 Test of Incrementing Number of Rooms by 1

## 5.1 Change of Schedules

Since the number of rooms is a constraint in this study, more rooms mean fewer postponed patients. That is why the aim of the maximum placement algorithm is to place the maximum number of patients in a limited number of rooms. This study also tests the differences in both algorithms when the number of rooms increases to 4.

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.8333 | 0.7500 | 0.7917 |
| 2 | 1 | 0.7917 | 0.6250 |
| 3 | 0.7500 | 0.6667 | 0.3750 |
| 4 | 0.8333 | 0.4167 | 0.2083 |
| 5 | 0.8750 | 0.9583 | 0.6667 |

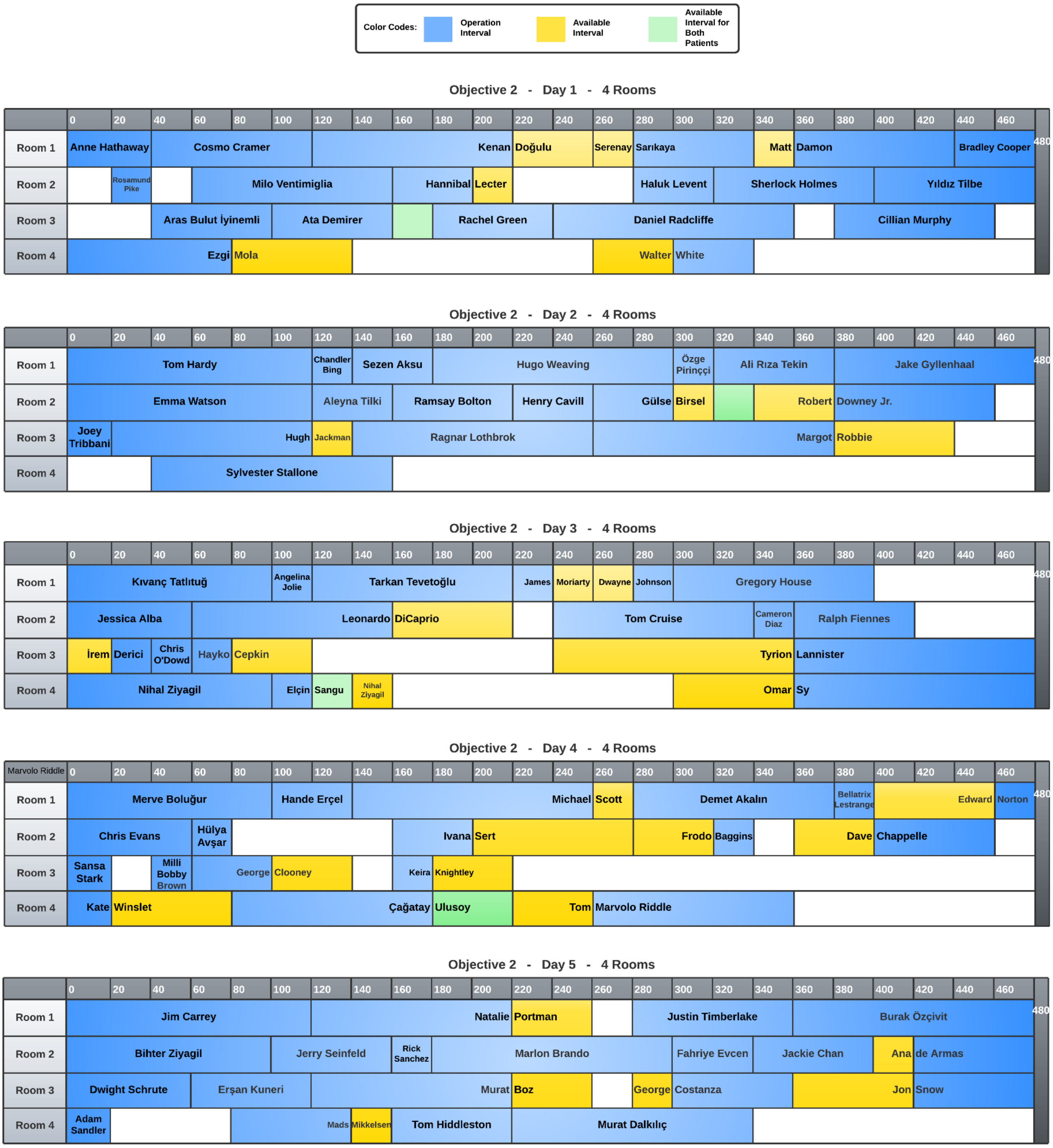
| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.8750 | 0.7500 | 0.7917 | 0.2500 |
| 2 | 1 | 0.7917 | 0.7500 | 0.2500 |
| 3 | 0.7500 | 0.7083 | 0.3750 | 0.5000 |
| 4 | 0.8333 | 0.4167 | 0.2083 | 0.4583 |
| 5 | 0.8750 | 0.9583 | 0.7083 | 0.5417 |

Above matrices are the utility matrices of the max placement objective. The left-hand-side matrix shows the case of 3 rooms, and the right-hand-side matrix shows the case of 4 rooms. They denote that by increasing the number of rooms by one, the schedule becomes more crowded, causing more usage of each room. Also, the number of postponed patients becomes less because of the existence of the 4th room.

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.8750 | 0.7917 | 0.8750 |
| 2 | 1 | 0.7917 | 0.7083 |
| 3 | 0.7500 | 0.8750 | 0.5000 |
| 4 | 0.7500 | 0.7917 | 0.5417 |
| 5 | 0.8750 | 0.9583 | 0.7083 |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.9583 | 0.9583 | 0.5833 | 0.1667 |
| 2 | 0.9583 | 0.7917 | 0.7917 | 0.2500 |
| 3 | 0.7500 | 0.7500 | 0.3750 | 0.4583 |
| 4 | 0.7500 | 0.7083 | 0.6667 | 0.2083 |
| 5 | 0.9583 | 0.8750 | 0.8333 | 0.3750 |

The above matrices are the utility matrices of the utility maximization objective. While the left-hand-side matrix shows the case of 3 rooms, the right-hand-side matrix shows us the case of 4 rooms. The density of each room also increases by increasing the number of rooms by one. Utilization of the first and the second room raises. Moreover, some of the postponed patients are carried to the 4th room. Therefore, it can be observed that the new schedule tends to satisfy the occurrence of more operations.




Gantt Chart of Maximum Placement Algorithm with 4 Rooms

## 5.1 Usage of Rooms

Utility Matrices of Maximum Patient Placement (Number of rooms are 3 and 4 respectively)

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.8333 | 0.7500 | 0.7917 |
| 2 | 1 | 0.7917 | 0.6250 |
| 3 | 0.7500 | 0.6667 | 0.3750 |
| 4 | 0.8333 | 0.4167 | 0.2083 |
| 5 | 0.8750 | 0.9583 | 0.6667 |

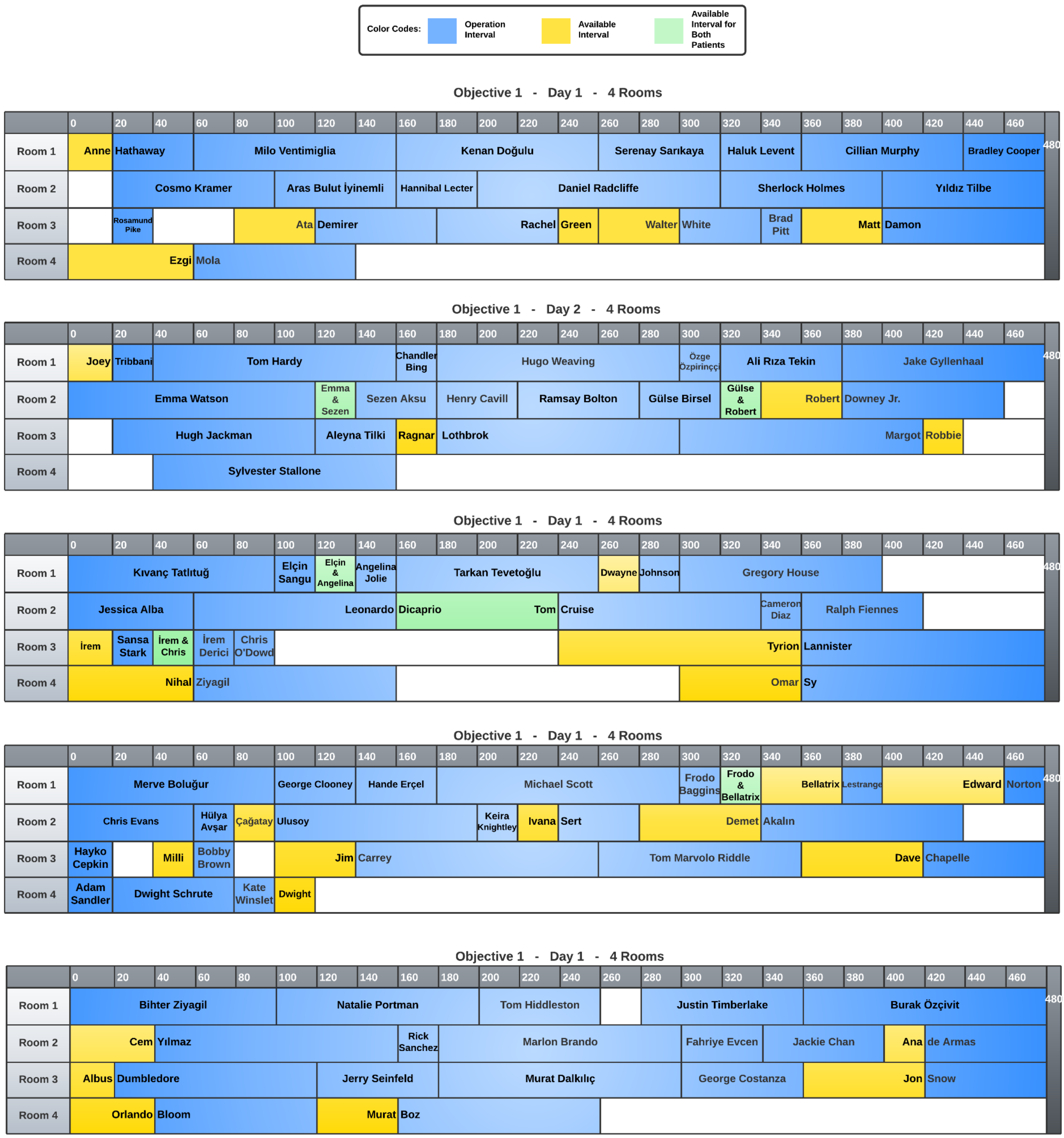
| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.8750 | 0.7500 | 0.7917 | 0.2500 |
| 2 | 1 | 0.7917 | 0.7500 | 0.2500 |
| 3 | 0.7500 | 0.7083 | 0.3750 | 0.5000 |
| 4 | 0.8333 | 0.4167 | 0.2083 | 0.4583 |
| 5 | 0.8750 | 0.9583 | 0.7083 | 0.5417 |

Utility Matrices of Maximum Room Utility (Number of rooms are 3 and 4 respectively)

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.8750 | 0.7917 | 0.8750 |
| 2 | 1 | 0.7917 | 0.7083 |
| 3 | 0.7500 | 0.8750 | 0.5000 |
| 4 | 0.7500 | 0.7917 | 0.5417 |
| 5 | 0.8750 | 0.9583 | 0.7083 |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.9583 | 0.9583 | 0.5833 | 0.1667 |
| 2 | 0.9583 | 0.7917 | 0.7917 | 0.2500 |
| 3 | 0.7500 | 0.7500 | 0.3750 | 0.4583 |
| 4 | 0.7500 | 0.7083 | 0.6667 | 0.2083 |
| 5 | 0.9583 | 0.8750 | 0.8333 | 0.3750 |

When the two cases are observed instantaneously, it can be concluded that the usage of rooms tends to be not fair; that is, the first and second rooms are clearly being used more than the others. Especially, the fourth room does not fit the fairness of the room's validity. It has less usage than the others in a significant way.




Gantt Chart of Maximum Utility Algorithm with 4 Rooms

# 7 Conclusion

To conclude, the scheduling problem’s perspectives have been observed. The general logic of priority and duration constraints has been understood. By changing the constraint’s values, the shadow prices also have been explained. Thanks to the problem, the scheduling vision has been gained.