

# Operation Scheduler



Group members: Stef Blom, Amin Suripin, Min Chen, Brian Tromp

Supervisor : Dennis Roubos

# Problem formulation

In hospitals there are operation rooms in which operations are carried out. Each operation has to be assigned to an operation room in such a way that the objective is being optimized.

We have build a Decision Support System (DSS) to determine the smartest schedule for a day and the corresponding performance, given the following inputs.

- An objective
- A list of operations that need to be carried out
- Restrictions
- Historical data (optional)

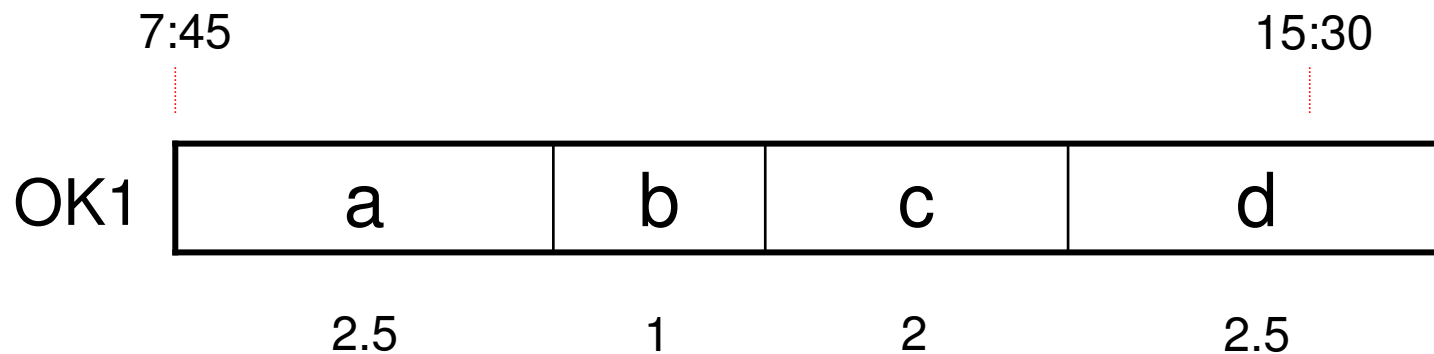
# Different objectives (1)

- Minimize overtime with a given occupancy
- Minimize the number of cancellations with a given occupancy
- Maximize the number of planned operations

# Different objectives (2)

- Minimize overtime with a given occupancy

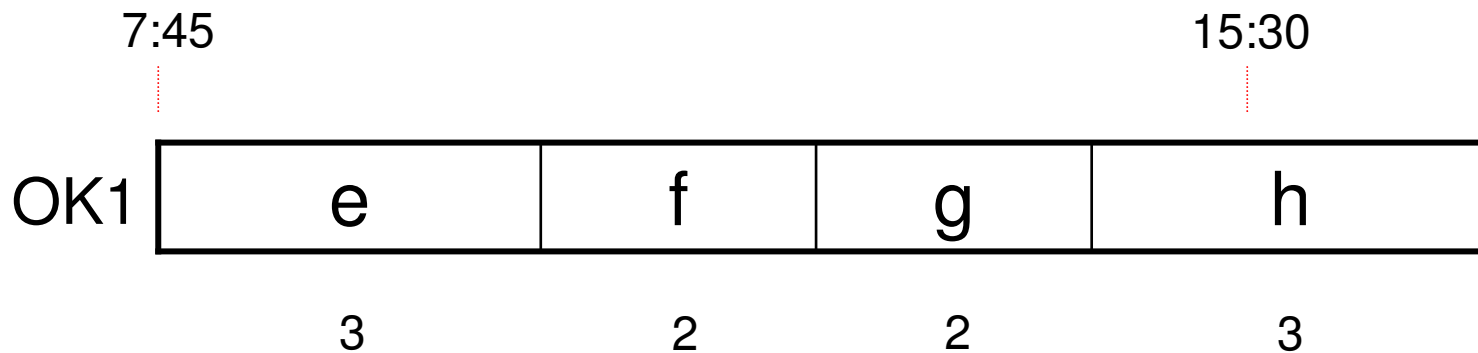
$$P(\text{Overtime}) = P(\text{Duration of operations} > 7.45 \text{ hrs})$$



# Different objectives (3)

- Minimize the number of cancellations

Cancel if  $P(\text{Overtime}) > 50\%$



# Different objectives (4)

- Maximize the number of planned operations

# of planned operations – expected # of cancellations

# List of operations



# Restrictions

- Working day
- Time between operations
- Maximal expected overtime



# Solution technique

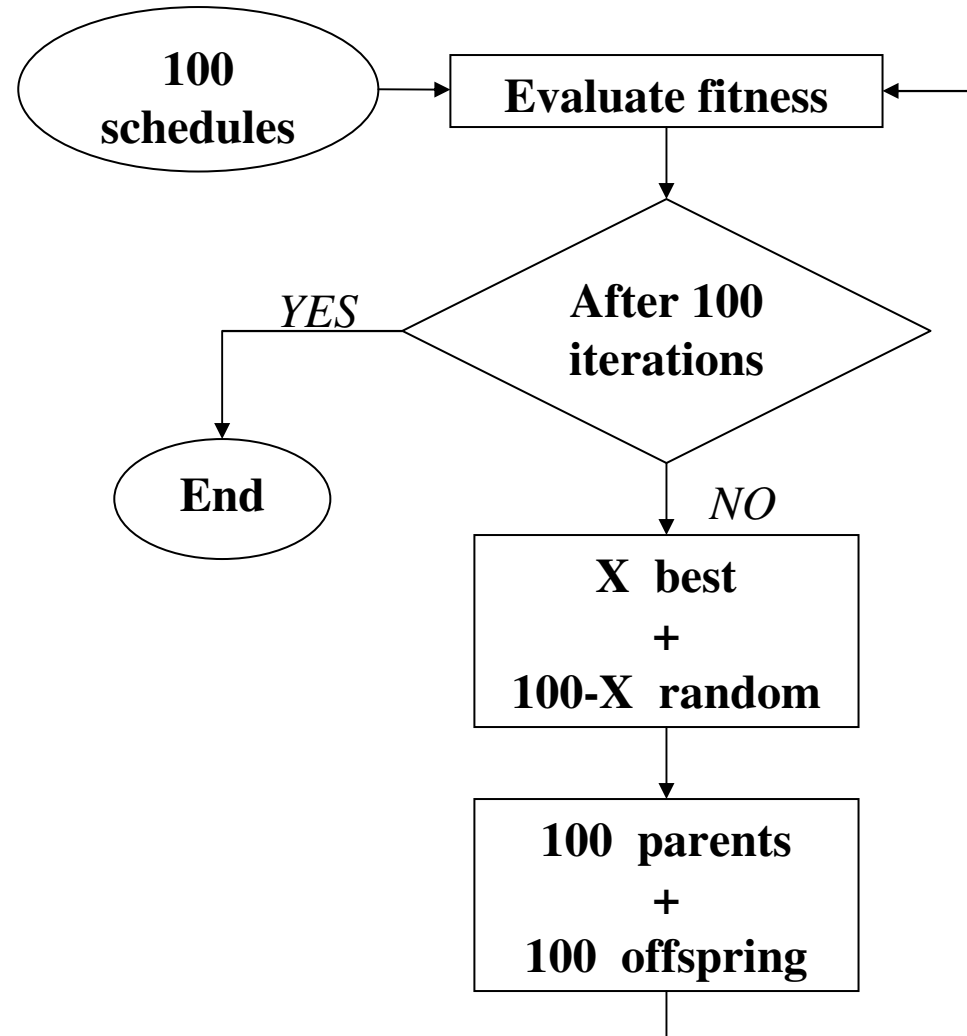
A Genetic Algorithm (GA) is a search technique to find the exact or approximate solutions to optimization problems.

Properties of GA:

- inheritance
- mutation
- selection
- crossover.

# Used GA

- **Step 0:**  
Create a random initial population
- **Step 1:**  
Evaluate the fitness of each individual in the population
- **Step 2:**  
Stop if 100 iterations have been performed.  
Else, move to step 3
- **Step 3:**  
Selecting individuals according to their fitness and randomly
- **Step 4:**  
Applying genetic operations to chosen parents to generate offspring



# Applying genetic operations (1)

- Remove operation
- Add operation
- Reschedule operation
- Change operation order in a room

# Applying genetic operations (2)

## Initial schedule

ID:	1	2	3	4	5	6	7	8	9	10	11	12
OK1:	1	1	0	0	0	0	0	0	0	1	0	0
OK2:	0	0	0	1	0	0	0	1	0	0	1	0
OK3:	0	0	1	0	1	1	0	0	0	0	0	0
Facility:	1	1	1	1	1	1	0	1	0	1	1	0

# Applying genetic operations (3)

## Remove operation

ID:	1	2	3	4	5	6	7	8	9	10	11	12
OK1:	1	1	0	0	0	0	0	0	0	1	0	0
OK2:	0	0	0	1	0	0	0	1	0	0	1	0
OK3:	0	0	0	0	1	1	0	0	0	0	0	0
Facility:	1	1	0	1	1	1	0	1	0	1	1	0

# Applying genetic operations (4)

## Add operation

ID:	1	2	3	4	5	6	7	8	9	10	11	12
OK1:	1	1	0	0	0	0	0	0	0	1	0	0
OK2:	0	0	0	1	0	0	1	1	0	0	1	0
OK3:	0	0	0	0	1	1	0	0	0	0	0	0
Facility:	1	1	0	1	1	1	1	1	0	1	1	0

# Applying genetic operations (5)

## Reschedule operation

ID:	1	2	3	4	5	6	7	8	9	10	11	12
OK1:	1	1	0	1	0	0	0	0	0	1	0	0
OK2:	0	0	0	0	0	0	1	1	0	0	1	0
OK3:	0	0	0	0	1	1	0	0	0	0	0	0
Facility:	1	1	0	1	1	1	1	1	0	1	1	0

# Applying genetic operations (6)

Change operation orders in room

ID:	10	2	3	4	5	6	7	8	9	1	11	12
OK1:	1	1	0	1	0	0	0	0	0	1	0	0
OK2:	0	0	0	0	0	0	1	1	0	0	1	0
OK3:	0	0	0	0	1	1	0	0	0	0	0	0
Facility:	1	1	0	1	1	1	1	1	0	1	1	0

# Result

## Schedule to be carried out

ID:	1	2	3	4	5	6	7	8	9	10	11	12
OK1:	1	1	0	1	0	0	0	0	0	1	0	0
OK2:	0	0	0	0	0	0	1	1	0	0	1	0
OK3:	0	0	0	0	1	1	0	0	0	0	0	0
Facility:	1	1	0	1	1	1	1	1	0	1	1	0

# Demo

**Operation Scheduler** [X]

**Objective**

Minimize Overtime      Minimum Occupancy:  %

Minimize Cancellations      Don't schedule if the probability of cancellation is more than:  %

Maximize # of Operations

**Add Room**

Name:

Type(s):

**Import / Export Rooms**

**Working Day**

Begin:

End:

**Time Between Operations**

Minutes

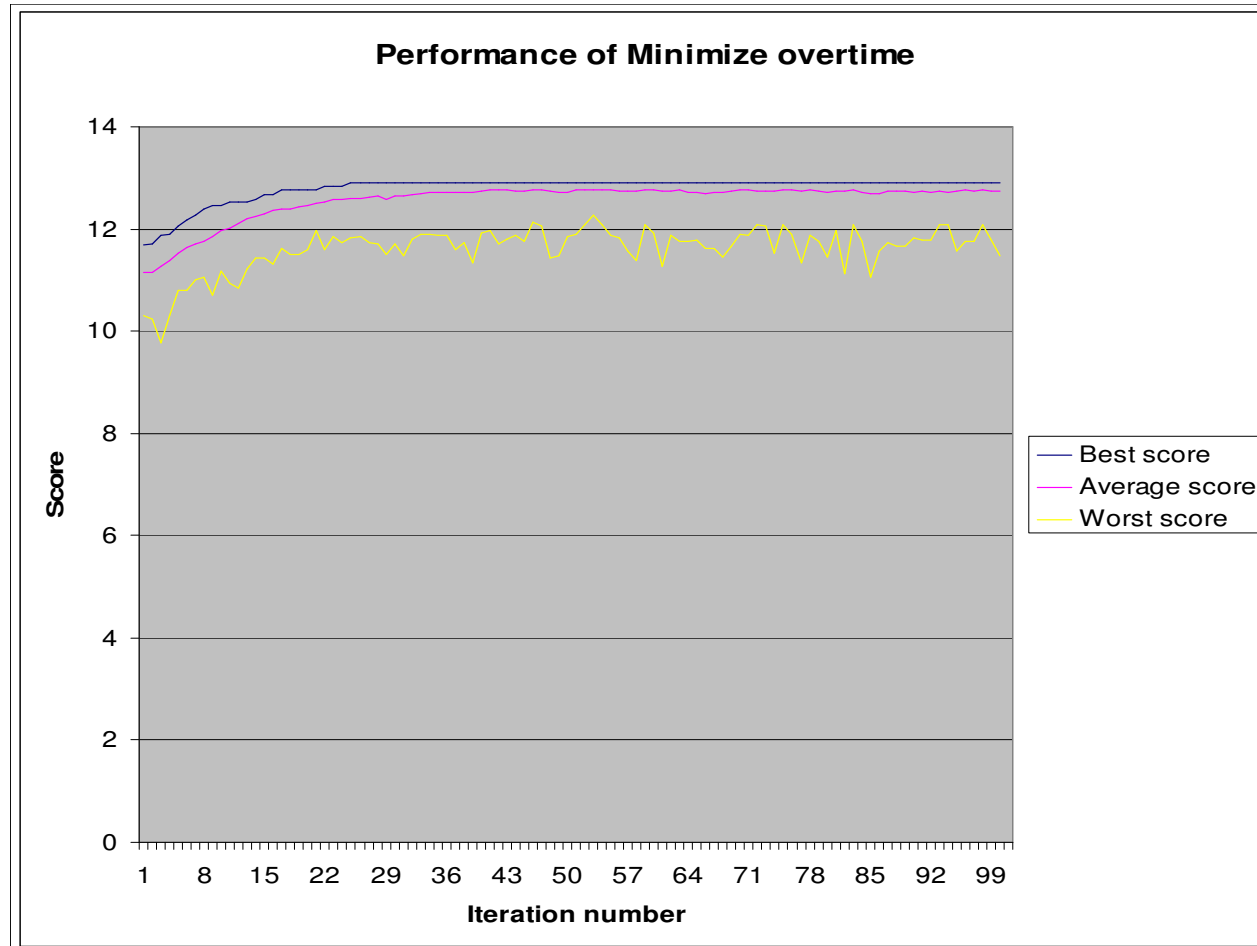
**Cancellation Constraint**

Cancel operation when expected overtime is more than  minutes.

**Rooms**

Name:	Type(s):
OK1	GYN;NCH;URO;VAT
OK2	GYN;HLO;KIN;NCH;URK;URO;VAT
OK3	GYN;HLO;HON;NCH;URK
OK4	GYN;HLO;NCH;URK;URO
OK7	GYN;HLO;NCH;TRA;VAT
OK8	GYN
OK9	GYN;HLO;NCH;VAT
OK10	HLO;NCH;VAT
OK11	GYN;HEG;HLO;HON;URK;VAT
OK12	GYN;URK;URO
OK13	NCH
OK14	NCH
OK15	HLO;NCH;VAT
OK17	NCH
OK18	NCH

# Performance



# Questions

