

# 1. Doodle Scheduling [10 pts]

by Roumen Guha, on Sunday, February 19th, 2017

Doodle Inc. is looking to interview a candidate for a new software engineer position at their company. It works like this: the interview (10 AM to 3 PM) is divided into a number of 20-minute time slots that may be used for 1-on-1 meetings with the candidate. There is also a one-hour time slot in the middle of the day where 3 employees take the candidate out for lunch. It would be nice for all 15 senior employees to meet with the candidate at some point during the day, but everybody has a busy schedule so it's not clear whether this will be possible. A doodle poll (obviously) was sent to the 15 senior employees to figure out their availability.

|         | 10:00 | 10:20 | 10:40 | 11:00 | 11:20 | 11:40 | Lunch | 1:00 | 1:20 | 1:40 | 2:00 | 2:20 | 2:40 |
|---------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|
| Manuel  | 0     | 0     | 1     | 1     | 0     | 0     | 0     | 1    | 1    | 0    | 0    | 0    | 0    |
| Luca    | 0     | 1     | 1     | 0     | 0     | 0     | 0     | 0    | 1    | 1    | 0    | 0    | 0    |
| Jule    | 0     | 0     | 0     | 1     | 1     | 0     | 1     | 1    | 0    | 1    | 1    | 1    | 1    |
| Michael | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1    | 1    | 1    | 1    | 1    | 0    |
| Malte   | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 1    | 1    | 0    | 0    | 0    | 0    |
| Chris   | 0     | 1     | 1     | 0     | 0     | 0     | 0     | 0    | 1    | 1    | 0    | 0    | 0    |
| Spyros  | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 0    | 0    | 0    | 0    | 0    | 0    |
| Mirjam  | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 1    | 1    | 1    |
| Matt    | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0    | 0    | 1    | 1    | 0    | 0    |
| Florian | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1    | 1    | 0    | 0    | 0    | 0    |
| Josep   | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 1    | 1    | 0    | 0    | 0    | 0    |
| Joel    | 1     | 1     | 0     | 0     | 0     | 1     | 1     | 1    | 1    | 0    | 0    | 1    | 1    |
| Tom     | 1     | 1     | 1     | 0     | 1     | 1     | 0     | 0    | 0    | 0    | 0    | 1    | 1    |
| Daniel  | 0     | 1     | 1     | 1     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    |
| Anne    | 1     | 1     | 0     | 0     | 1     | 1     | 0     | 0    | 0    | 0    | 0    | 0    | 0    |

In the table, a 1 means that the employee is available at the indicated time, while a 0 means that they are unavailable. Determine whether a feasible interview schedule exists. If so, print out a calendar for the candidate that lists who they will be meeting at each time slot.



```
In [7]: using JuMP

m = Model()

@variable(m, x[names, times] >= 0)

# the candidate can only meet with 1 employee at any 20-minute period
@constraint(m, a[j in times], sum(x[i,j] for i in names) <= 1)

# each employee can only meet with the candidate once
@constraint(m, b[i in names], sum(x[i,j] for j in times) <= 1)

@objective(m, Max, sum(x[i,j]*data[i,j] for i in names, j in times))

status = solve(m)
println(status)

schedule = NamedArray( [Int(getvalue(x[i,j])) for i in names, j in times], (names, times), ("Names", "Times"))

show(IOContext(STDOUT, displaysize=(100, 1000)), schedule)
```

Optimal  
15×15 Named Array{Int64,2}

| Names \ Times | 10:00 | 10:20 | 10:40 | 11:00 | 11:20 | 11:40 | 12:00 | 12:20 | 12:40 | 13:00 | 13:20 | 13:40 | 14:00 | 14:20 | 14:40 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Manuel        | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Luca          | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Jule          | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Michael       | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Malte         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |
| Chris         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| Spyros        | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Mirjam        | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     |
| Matt          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     |
| Florian       | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     |
| Josep         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     |
| Joel          | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Tom           | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     |
| Daniel        | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Anne          | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Note:**

To account for the lunch period, I simply split the lunch hour into 3 separate 20-minute periods with identical data.

```
In [9]: println("Schedule is as follows: ")
println()

for j = 1:15
    for i = 1:15
        if schedule[i,j] == 1
            print(times[j])
            print(": ")
            println(names[i])
        end
    end
end
```

Schedule is as follows:

```
10:00: Joel
10:20: Luca
10:40: Manuel
11:00: Daniel
11:20: Spyros
11:40: Anne
12:00: Jule
12:20: Michael
12:40: Malte
13:00: Josep
13:20: Florian
13:40: Chris
14:00: Matt
14:20: Mirjam
14:40: Tom
```

This means that Jule, Michael and Malte get to join the candidate for lunch. Lucky them.

```
In [11]: println(m)
```

```

Max [Manuel,10:40] + x[Manuel,11:00] + x[Manuel,13:00] + x[Manuel,13:20] + x[Luca,10:20] + x[Luca,10:40] + x[Luca,13:20] + x[Luca,13:40] + x[Julie,11:00] + x[Julie,11:20] + x[Julie,12:00] + x[Julie,12:20] + x[Julie,12:40] + x[Julie,13:00] + x[Julie,13:40] + x[Julie,14:00] + x[Julie,14:20] + x[Julie,14:40] + x[Michael,11:00] + x[Michael,11:20] + x[Michael,11:40] + x[Michael,12:00] + x[Michael,12:20] + x[Michael,12:40] + x[Michael,13:00] + x[Michael,13:20] + x[Michael,13:40] + x[Michael,14:00] + x[Michael,14:20] + x[Malte,12:00] + x[Malte,12:20] + x[Malte,12:40] + x[Malte,13:00] + x[Malte,13:20] + x[Chris,10:20] + x[Chris,10:40] + x[Chris,13:20] + x[Chris,13:40] + x[Spyros,11:00] + x[Spyros,11:20] + x[Spyros,11:40] + x[Spyros,12:00] + x[Spyros,12:20] + x[Spyros,12:40] + x[Mirjam,10:00] + x[Mirjam,10:20] + x[Mirjam,14:00] + x[Mirjam,14:20] + x[Mirjam,14:40] + x[Matt,10:00] + x[Matt,10:20] + x[Matt,10:40] + x[Matt,13:40] + x[Matt,14:00] + x[Florian,13:00] + x[Florian,13:20] + x[Josep,12:00] + x[Josep,12:20] + x[Josep,12:40] + x[Josep,13:00] + x[Josep,13:20] + x[Joel,10:00] + x[Joel,10:20] + x[Joel,11:40] + x[Joel,12:00] + x[Joel,12:20] + x[Joel,12:40] + x[Joel,13:00] + x[Joel,13:20] + x[Joel,14:20] + x[Joel,14:40] + x[Tom,10:00] + x[Tom,10:20] + x[Tom,10:40] + x[Tom,11:20] + x[Tom,11:40] + x[Tom,14:20] + x[Tom,14:40] + x[Daniel,10:20] + x[Daniel,10:40] + x[Daniel,11:00] + x[Anne,10:00] + x[Anne,10:20] + x[Anne,11:20] + x[Anne,11:40]

```

Subject to

[illegible]

```

n,13:00] + x[Florian,13:20] + x[Florian,13:40] + x[Florian,14:00] + x[Florian,14:20] + x[Florian,14:40] <= 1
x[Josep,10:00] + x[Josep,10:20] + x[Josep,10:40] + x[Josep,11:00] + x[Josep,11:20] + x[Josep,11:40] + x[Josep,12:00] + x[Josep,12:20] + x[Josep,12:40] + x[Josep,13:00] + x[Josep,13:20] + x[Josep,13:40] + x[Josep,14:00] + x[Josep,14:20] + x[Josep,14:40] <= 1
x[Joel,10:00] + x[Joel,10:20] + x[Joel,10:40] + x[Joel,11:00] + x[Joel,11:20] + x[Joel,11:40] + x[Joel,12:00] + x[Joel,12:20] + x[Joel,12:40] + x[Joel,13:00] + x[Joel,13:20] + x[Joel,13:40] + x[Joel,14:00] + x[Joel,14:20] + x[Joel,14:40] <= 1
x[Tom,10:00] + x[Tom,10:20] + x[Tom,10:40] + x[Tom,11:00] + x[Tom,11:20] + x[Tom,11:40] + x[Tom,12:00] + x[Tom,12:20] + x[Tom,12:40] + x[Tom,13:00] + x[Tom,13:20] + x[Tom,13:40] + x[Tom,14:00] + x[Tom,14:20] + x[Tom,14:40] <= 1
x[Daniel,10:00] + x[Daniel,10:20] + x[Daniel,10:40] + x[Daniel,11:00] + x[Daniel,11:20] + x[Daniel,11:40] + x[Daniel,12:00] + x[Daniel,12:20] + x[Daniel,12:40] + x[Daniel,13:00] + x[Daniel,13:20] + x[Daniel,13:40] + x[Daniel,14:00] + x[Daniel,14:20] + x[Daniel,14:40] <= 1
x[Anne,10:00] + x[Anne,10:20] + x[Anne,10:40] + x[Anne,11:00] + x[Anne,11:20] + x[Anne,11:40] + x[Anne,12:00] + x[Anne,12:20] + x[Anne,12:40] + x[Anne,13:00] + x[Anne,13:20] + x[Anne,13:40] + x[Anne,14:00] + x[Anne,14:20] + x[Anne,14:40] <= 1
x[i,j] >= 0 for all i in {Manuel,Luca,...,Daniel,Anne}, j in {10:00,10:20,...,14:20,14:40}

```