COMP SCI 524 - Homework 3

Jingde Wan

Q1

In [12]:

```
using JuMP, NamedArrays

availability =
  [ 0 0 1 1 0 0 0 1 1 0 0 0 0
    0 1 1 0 0 0 0 0 1 1 0 0 0
    0 0 0 1 1 0 1 1 0 1 1 1 1
    0 0 0 1 1 1 1 1 1 1 1 1 0
    0 0 0 0 0 0 1 1 1 0 0 0 0
    0 1 1 0 0 0 0 0 1 1 0 0 0
    0 0 0 1 1 1 1 0 0 0 0 0 0
    1 1 0 0 0 0 0 0 0 0 1 1 1
    1 1 1 0 0 0 0 0 0 1 1 0 0
    0 0 0 0 0 0 0 1 1 0 0 0 0
    0 0 0 0 0 0 1 1 1 0 0 0 0
    1 1 0 0 0 1 1 1 1 0 0 1 1
    1 1 1 0 1 1 0 0 0 0 0 1 1
    0 1 1 1 0 0 0 0 0 0 0 0 0
    1 1 0 0 1 1 0 0 0 0 0 0 0 ]

TIMES = ["10:00","10:20","10:40","11:00","11:20","11:40","lunch","1:00","1:20","1:40","2:00","2:
NAMES = [:Manuel,:Luca,:Jule,:Michael,:Malte,:Chris,:Spyros,:Mirjam,:Matt,:Florian,:Josep,:Joel
times = NamedArray( availability, (NAMES,TIMES), ("NAME","TIME"))
```

Out[12]:

```
15×13 Named Array{Int64,2}
NAME ╲ TIME │ 10:00  10:20  10:40  11:00  …   1:40   2:00   2:20   2:40
────────────┼──────────────────────────────────────────────────────────────
──────────────────────────────────
:Manuel     │     0      0      1      1  …      0      0      0      0
:Luca       │     0      1      1      0         1      0      0      0
:Jule       │     0      0      0      1         1      1      1      1
:Michael    │     0      0      0      1         1      1      1      0
:Malte      │     0      0      0      0         0      0      0      0
:Chris      │     0      1      1      0         1      0      0      0
:Spyros     │     0      0      0      1         0      0      0      0
:Mirjam     │     1      1      0      0         0      1      1      1
:Matt       │     1      1      1      0         1      1      0      0
:Florian    │     0      0      0      0         0      0      0      0
:Josep      │     0      0      0      0         0      0      0      0
:Joel       │     1      1      0      0         0      0      1      1
:Tom        │     1      1      1      0         0      0      1      1
:Daniel     │     0      1      1      1         0      0      0      0
:Anne       │     1      1      0      0  …      0      0      0      0
```

In [26]:

```
using JuMP, Clp

m = Model(Clp.Optimizer)

@variable(m, 0 <= x[1:15,1:13] <= 1)

# meet candidate at some time slot for each senior
for i in 1:15
    @constraint(m, sum(x[i,:]) == 1)
end

# only one senior at each time slot except for lunch
for j in 1:6
    @constraint(m, sum(x[:,j]) == 1)
end

@constraint(m, sum(x[:,7]) == 3)

for j in 8:13
```

```
    @constraint(m, sum(x[:, j]) == 1)
end

@objective(m, Max, sum(x[i, j]*times[i, j] for i in 1:15, j in 1:13))

optimize!(m)
println(termination_status(m))

assignment = NamedArray( [ (value.(x[i, j])) for i in 1:15, j in 1:13 ], (NAMES, TIMES), ("NAME"
println(assignment)
```

```
OPTIMAL
15×13 Named Array{Float64,2}
NAME ╲ TIME │ 10:00  10:20  10:40  11:00  …   1:40   2:00   2:20   2:40
────────────┼─────────────────────────────────────────────────────────────
────────────────────────────────────
:Manuel     │   0.0    0.0    0.0    0.0  …    0.0    0.0    0.0    0.0
:Luca       │   0.0    1.0    0.0    0.0       0.0    0.0    0.0    0.0
:Jule       │   0.0    0.0    0.0    0.0       0.0    0.0    0.0    1.0
:Michael    │   0.0    0.0    0.0    1.0       0.0    0.0    0.0    0.0
:Malte      │   0.0    0.0    0.0    0.0       0.0    0.0    0.0    0.0
:Chris      │   0.0    0.0    0.0    0.0       1.0    0.0    0.0    0.0
:Spyros     │   0.0    0.0    0.0    0.0       0.0    0.0    0.0    0.0
:Mirjam     │   1.0    0.0    0.0    0.0       0.0    0.0    0.0    0.0
:Matt       │   0.0    0.0    0.0    0.0       0.0    1.0    0.0    0.0
:Florian    │   0.0    0.0    0.0    0.0       0.0    0.0    0.0    0.0
:Josep      │   0.0    0.0    0.0    0.0       0.0    0.0    0.0    0.0
:Joel       │   0.0    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    1.0    0.0
:Daniel     │   0.0    0.0    1.0    0.0       0.0    0.0    0.0    0.0
:Anne       │   0.0    0.0    0.0    0.0  …    0.0    0.0    0.0    0.0
Coin0506I Presolve 28 (0) rows, 195 (0) columns and 390 (0) elements
Clp0006I 0  Obj 0 Primal inf 29.999997 (28) Dual inf 72.999993 (73)
Clp0006I 41  Obj 22 Primal inf 54.999998 (24)
Clp0006I 72  Obj 15 Primal inf 0.9999999 (1)
Clp0006I 73  Obj 15
Clp0000I Optimal - objective value 15
Clp0032I Optimal objective 15 - 73 iterations time 0.002
```

Q2

In this problem, I need to determine the internal transfer of cars among 10 agencies. $x[i, j]$ denotes the number of cars transferred from agency i to agency j, where $i = 2, 5, 8, 9$, $j = 1, 3, 4, 6, 7, 10$.

In [46]:

```
using JuMP, Clp


m1 = Model(Clp.Optimizer)

cars_in = [1,3,4,6,7,10]
cars_out = [2,5,8,9]

xcoord = [0 20 18 30 35 33 5 5 11 2]
ycoord = [0 20 10 12 0 25 27 10 0 15]

@variable(m1, x[1:4, 1:6] >= 0)

# cars in
@constraint(m1, sum(x[:,1]) >= 2) #j=1
@constraint(m1, sum(x[:,2]) >= 4) #j=3
@constraint(m1, sum(x[:,3]) >= 3) #j=4
@constraint(m1, sum(x[:,4]) >= 5) #j=6
@constraint(m1, sum(x[:,5]) >= 1) #j=7
@constraint(m1, sum(x[:,6]) >= 5) #j=10

#cars out
@constraint(m1, sum(x[1,:]) <= 7) #i=2
@constraint(m1, sum(x[2,:]) <= 3) #i=5
@constraint(m1, sum(x[3,:]) <= 4) #i=8
@constraint(m1, sum(x[4,:]) <= 7) #i=9
```

```
@objective(m1, Min, sum( 0.5*x[i,j]*1.3*sqrt((xcoord[cars_out[i]]-xcoord[cars_in[j]])^2
            +(ycoord[cars_out[i]]-ycoord[cars_in[j]])^2) for i in 1:4, j in 1:6))

optimize!(m1)
println("The total minimum cost incurred for transport: ", objective_value(m1))
println("The movement is shown below (i is the vertical index, j is the horizontal index):")
for i in 1:4
    for j in 1:6
        print(getvalue(x[i,j]), "   ")
    end
    println()
end
```

```
The total minimum cost incurred for transport: 152.63901632295628
The movement is shown below (i is the vertical index, j is the horizontal index):
0.0   1.0   0.0   5.0   1.0   0.0
0.0   0.0   3.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0   0.0   4.0
2.0   3.0   0.0   0.0   0.0   1.0
Coin0506I Presolve 10 (0) rows, 24 (0) columns and 48 (0) elements
Clp0006I 0  Obj 0 Primal inf 19.999999 (6)
Clp0006I 10  Obj 152.63902
Clp0000I Optimal - objective value 152.63902
Clp0032I Optimal objective 152.6390163 - 10 iterations time 0.002
```

Q3

(a)

In [47]:

```
tasks = 1:18
durations = [2 16 9 8 10 6 2 2 9 5 3 2 1 7 4 3 9 1]
predecessors = ( [], [1], [2], [2], [3], [4,5], [4], [6], [4,6], [4], [6], [9], [7], [2], [4,14],
pred_dict = Dict(zip(tasks,predecessors));   # dictionary mapping tasks --> predecessors.
```

Out[47]:

```
Dict{Int64,Array{T,1} where T} with 18 entries:
  18 => [17]
  2  => [1]
  16 => [8, 11, 14]
  11 => [6]
  7  => [4]
  9  => [4, 6]
  10 => [4]
  17 => [12]
  8  => [6]
  6  => [4, 5]
  4  => [2]
  3  => [2]
  5  => [3]
  13 => [7]
  14 => [2]
  15 => [4, 14]
  12 => [9]
  1  => Any[]
```

In [60]:

```
using JuMP, Clp
m2 = Model(Clp.Optimizer)

@variable(m2, tstart[tasks]>=0)

for i in tasks
    for j in pred_dict[i]
        @constraint(m2, tstart[i] >= tstart[j] + durations[j])
    end
end
@constraint(m2, tstart[1] == 0)
@objective(m2, Min, tstart[18] + durations[18])

optimize!(m2)
println("Earliest possible completion date: ", getobjectivevalue(m2))
for i in tasks
```

```
    println("Task ", i, " date of completion: ", getvalue(tstart[i]))
end
```

```
Earliest possible completion date: 64.0
Task 1 date of completion: 0.0
Task 2 date of completion: 2.0
Task 3 date of completion: 18.0
Task 4 date of completion: 18.0
Task 5 date of completion: 27.0
Task 6 date of completion: 37.0
Task 7 date of completion: 26.0
Task 8 date of completion: 43.0
Task 9 date of completion: 43.0
Task 10 date of completion: 26.0
Task 11 date of completion: 43.0
Task 12 date of completion: 52.0
Task 13 date of completion: 28.0
Task 14 date of completion: 18.0
Task 15 date of completion: 26.0
Task 16 date of completion: 46.0
Task 17 date of completion: 54.0
Task 18 date of completion: 63.0
Coin0506I Presolve 0 (-23) rows, 0 (-18) columns and 0 (-45) elements
Clp3002W Empty problem - 0 rows, 0 columns and 0 elements
Clp0000I Optimal - objective value 64
Coin0511I After Postsolve, objective 64, infeasibilities - dual 0 (0), primal 0 (0)
Clp0032I Optimal objective 64 - 0 iterations time 0.002, Presolve 0.00
```

(b)

In [67]:

```
# additional columns of data (maximum reduction possible )
max_reduction =  [0,  3,  1,  2,  2,  1, 1, 0,  2,  1,  1, 0, 0,  2,  2, 1,  3, 0]  # max reducti
cost_reduction = [0, 30, 26, 12, 17, 15, 8, 0, 42, 21, 18, 0, 0, 22, 12, 6, 16, 0]  # cost of re
bonus_amount = 30     # bonus for expediting the project ($1,000/week )
```

Out[67]:

```
30
```

In [68]:

```
using JuMP, Clp
m3 = Model(Clp.Optimizer)

@variable(m3, tstart[tasks] >= 0)
@variable(m3, treduct[tasks] >= 0)

for i in tasks
    @constraint(m3, treduct[i] <= max_reduction[i])
end

for i in tasks
    for j in pred_dict[i]
        @constraint(m3, tstart[i] >= tstart[j] + durations[j] - treduct[i])
    end
end
@constraint(m3, tstart[1] == 0)
@objective(m3, Max, bonus_amount*(64-tstart[18]-durations[18])-sum(cost_reduction[i]*treduct[i]

optimize!(m3)
println("Maximum profit: ", getobjectivevalue(m3))
println("Earliest date of completion: ", getvalue(tstart[18]) + durations[18])
for i in tasks
    println("Task ", i, " date of completion: ", getvalue(tstart[i]))
end
```

```
Maximum profit: 87.0
Earliest date of completion: 57.0
Task 1 date of completion: 0.0
Task 2 date of completion: 2.0
Task 3 date of completion: 17.0
Task 4 date of completion: 18.0
Task 5 date of completion: 24.0
Task 6 date of completion: 33.0
Task 7 date of completion: 26.0
Task 8 date of completion: 39.0
Task 9 date of completion: 39.0
Task 10 date of completion: 26.0
Task 11 date of completion: 39.0
Task 12 date of completion: 48.0
Task 13 date of completion: 28.0
Task 14 date of completion: 18.0
Task 15 date of completion: 26.0
Task 16 date of completion: 42.0
Task 17 date of completion: 47.0
Task 18 date of completion: 56.0
Coin0506I Presolve 5 (-36) rows, 9 (-27) columns and 18 (-67) elements
Clp0006I 0  Obj 273 Primal inf 11.199997 (3)
Clp0006I 3  Obj 87
Clp0000I Optimal - objective value 87
Coin0511I After Postsolve, objective 87, infeasibilities - dual 0 (0), primal 0 (0)
Clp0032I Optimal objective 87 - 3 iterations time 0.002, Presolve 0.00
```