

최적화이론

Term Project

선거운동 스케줄링 최적화 (Team8 #5)

산업경영공학부

2014170810

김기호

▶ #2를 그대로 적용하기에는 몇 가지 개선점이 발견되었고, #3에서의 해 찾기 방식은 Gurobi나 여러 가지 Heuristic 방법을 적용하기에 Robust 하지 않아, #2와 #3을 참고하여 새롭게 Formulation을 하고 해를 찾았습니다.

1. Decision Variables

X_{ij} : Whether to visit Person i on day j

M_j : Whether to have a meeting on day j

S_i : Whether having gained support of Person i ($= \text{Max}(X_{ij} - P_i + 1, 0)$)

B_i : Binary coefficient of Big M

X_{ij}, M_j, S_i, B_i : Binary Variable

2. Assumption

- 1) 하루에 한 명의 사람을 여러 번 만날 수 없다.
- 2) 지지를 얻고 나면 더 이상 만나지 않는다.
- 3) 휴일 및 공휴일에는 선거운동을 하지 않는다.
- 4) 일주일에 하루는 회의에 참석해야하며 이 때, 선거운동을 하지 못한다.

3. Parameter

- 1) P_j : Support which can be gained from Person i

& Visiting needed to gain support of Person i

등급	인물코드	필요방문횟수
A	01	8
A	02	8
A	03	8
B	04	6
B	05	6
B	06	6
B	07	6
B	08	6
B	09	6
B	10	6
B	11	6
B	12	6
B	13	6
C	14	4
C	15	4
C	16	4
C	17	4
C	18	4
C	19	4
C	20	4
C	21	4
C	22	4

2) C_{ij} : Cost by each visiting (=1.5 for each)

기존의 목적함수인 $\sum_i^{22}(P_i \times S_i)$ 의 결과는 아래와 같습니다.

	일																															기준	충족	지지도	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	방문	기준	충족	지지도
인사	1	0	0	1	1	1	0	1	0	0	1	0	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	8	8	TRUE	8
	2	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	8	8	TRUE	8
	3	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	8	8	TRUE	8
	4	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	TRUE	6
	5	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	6	6	TRUE	6
	6	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	6	6	TRUE	6
	7	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	6	6	TRUE	6
	8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	6	6	TRUE	6
	9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	6	6	TRUE	6
	10	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	6	6	TRUE	6
	11	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	6	6	TRUE	6
	12	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	6	6	TRUE	6
	13	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	6	6	TRUE	6	
	14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	6	FALSE	0
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	4	FALSE	0	
	16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	FALSE	0
	17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4	TRUE	4
	18	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	FALSE	0
	19	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	4	4	TRUE	4
	20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	4	4	TRUE	4
	21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	4	4	TRUE	4
	22	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	4	4	TRUE	4
방문		6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
회의		0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0			104	

(1) 위와 같이 기존의 목적함수대로면 매일 6번을 꼭 채워서 방문하게 되어

$\sum_i^{31} X_{ij} \geq 4$ for all j 제약조건이 의미가 없게 됩니다.

(2) 따라서 매번 방문하는 것에 대한 Cost를 일괄적으로 적용하였습니다.

(3) 그 크기를 -1로 해보았더니 얻은 총 지지도는 여전히 104였고, -2로 해보았더니 42로 크게 줄어드는 결과를 보여, 그 크기를 -1.5로 정하였습니다.

4. Objective Function & Constraints

$$\text{Maximize. } \sum_i^{22} (P_i \times S_i) - (X_{ij} \times C_{ij} (= 1.5))$$

Subject to.

Weekend Constraints

$$X_{ij} = 0 \quad \text{for } (i, j | j = 5, 6, 12, 13, 19, 20, 26, 27)$$

$$M_j = 0 \quad \text{for } (i, j | j = 5, 6, 12, 13, 19, 20, 26, 27)$$

Person02 Constraints

$$X_{2j} = 0 \quad \text{for } j = 3, 10, 17, 24, 31$$

Meeting Constraints

$$\sum_{j=1}^5 M_j = 1, \quad \sum_{j=6}^{12} M_j = 1, \quad \sum_{j=13}^{19} M_j = 1, \quad \sum_{j=20}^{26} M_j = 1, \quad \sum_{j=27}^{31} M_j = 1$$

$$M_j + X_{ij} \leq 1 \quad \text{for all } (i, j) - \text{Mutually exclusive rule between campaign and meeting}$$

$S_i = \text{Max}(X_{ij} - P_i + 1, 0)$ Constraints

$$S_i \geq \sum_j^{31} X_{ij} - P_i + 1 \quad \text{for all } i$$

$$S_i \geq 0 \quad \text{for all } i$$

$$S_i \leq \sum_j^{31} X_{ij} - P_i + 1 + MB_i \quad \text{for all } i$$

$$S_i \leq M(1 - B_i) \quad \text{for all } i$$

Assumption Constraints

$$\sum_j^{22} X_{ij} \leq P_i \quad \text{for all } i$$

Capacity Constraints

$$\sum_i^{31} X_{ij} \leq 6 \quad \text{for all } j$$

$$4 \times M_j + \sum_i^{31} X_{ij} \geq 4 \quad \text{for all } j$$

5. Source code

```
from gurobipy import *

# Parameters
penalty=-1.5
profit = (8,8,8,6,6,6,6,6,6,6,6,6,4,4,4,4,4,4,4,4)
candidate={(i) : profit[i] for i in range(22)}
day={(j) : 1 for j in range(31)}
visit={(i,j) : 1 for i in range(22) for j in range(31)}

# Model
m=Model()
m.getObjective()
m.setAttr('modelsense',0)

# Add variables
s=m.addVars(candidate.keys(), obj=profit, vtype=GRB.BINARY, name='s')
d=m.addVars(day.keys(), vtype=GRB.BINARY, name='d')
x=m.addVars(visit.keys() , obj=penalty,vtype=GRB.BINARY, name='x')
b=m.addVars(candidate.keys() , vtype=GRB.BINARY, name='b')

# Add constraints

#Weekend constraints
weekend={4,5,11,12,18,19,25,26}
notweekend={0,1,2,3,6,7,8,9,10,13,14,15,16,17,20,21,22,23,24,27,28,29,30}
constr1=m.addConstrs(x.sum('*',j) == 0 for j in weekend)
constr10=m.addConstrs(d.sum(j) == 0 for j in weekend)

#Person02 constraints
person02={2,9,16,23,30}
constr2=m.addConstr(x[1,2]+x[1,9]+x[1,16]+x[1,23]+x[1,30] == 0)

#Meeting constraints
stop=[5,12,19,26,31]
result=[[[]],[[]],[[]],[[]],[[]]]
for i in range(31):
    for j in range(5):
        if( (j==0 or i>=stop[j-1]) and i < stop[j] ) :
            result[j].append(d[i])
```

```

constr3=m.addConstr(sum(result[0]) == 1)
constr4=m.addConstr(sum(result[1]) == 1)
constr5=m.addConstr(sum(result[2]) == 1)
constr6=m.addConstr(sum(result[3]) == 1)
constr7=m.addConstr(sum(result[4]) == 1)
constr8=m.addConstrs(d.sum(j)+x.sum(i,j) <= 1 for i in range(len(candidate)) for j in
range(len(day)))

```

#Assumption

```

constr11=m.addConstrs(x.sum(i,'*') <= profit[i] for i in range (22))

```

#Max variable Constraints

```

constr12=m.addConstrs(s.sum(i) >= 0 for i in range (22))
constr13=m.addConstrs(s.sum(i) >= x.sum(i,'*')-profit[i]+1 for i in range (22))
constr14=m.addConstrs(s.sum(i) <= 2*(1-b[i]) for i in range (22))
constr15=m.addConstrs(s.sum(i) <= x.sum(i,'*')-profit[i]+1+2*b[i] for i in range
(22))

```

#Capacity Constraints

```

constr16=m.addConstrs(x.sum('*',j) <= 6 for j in range(31))
constr17=m.addConstrs(4*d.sum(j)+x.sum('*',j) >= 4 for j in notweekend)

```

```

m.update()
print("\n{0:{1} ^ 40}".format("Model","="))

```

```

m.display()
print("\n{0:{1} ^ 40}".format("", "="))

```

```

m.optimize()
print("\n{0:{1} ^ 40}".format("", "="))
print("\n{0:{1} ^ 40}".format("Variables Value", "="))

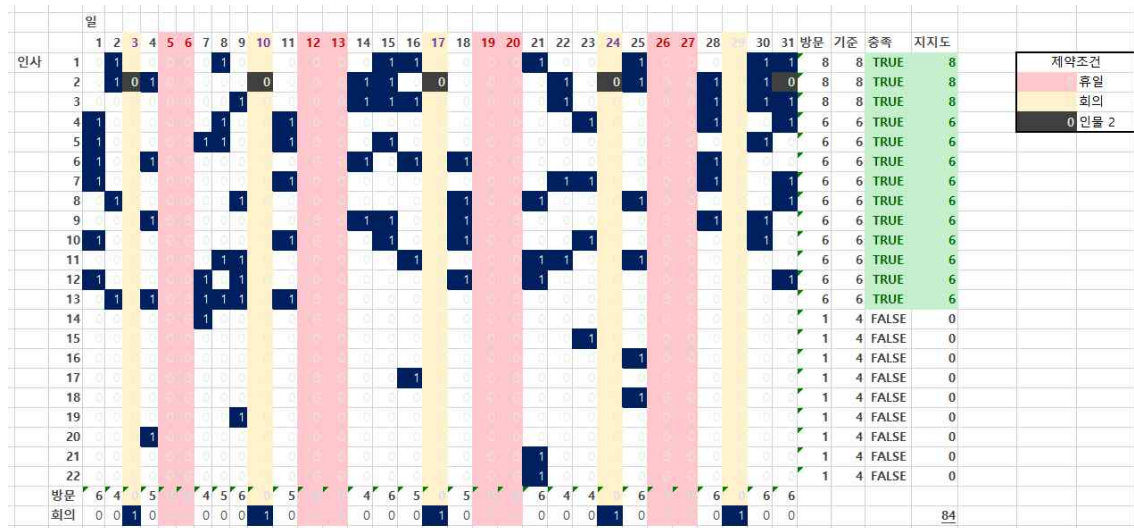
```

```

for v in m.getVars():
    print("{0} = {1:>277}".format(v.varName,v.x))
print("\n{0:{1} ^ 40}".format("", "="))

```

6. Solution



Optimal value = 84

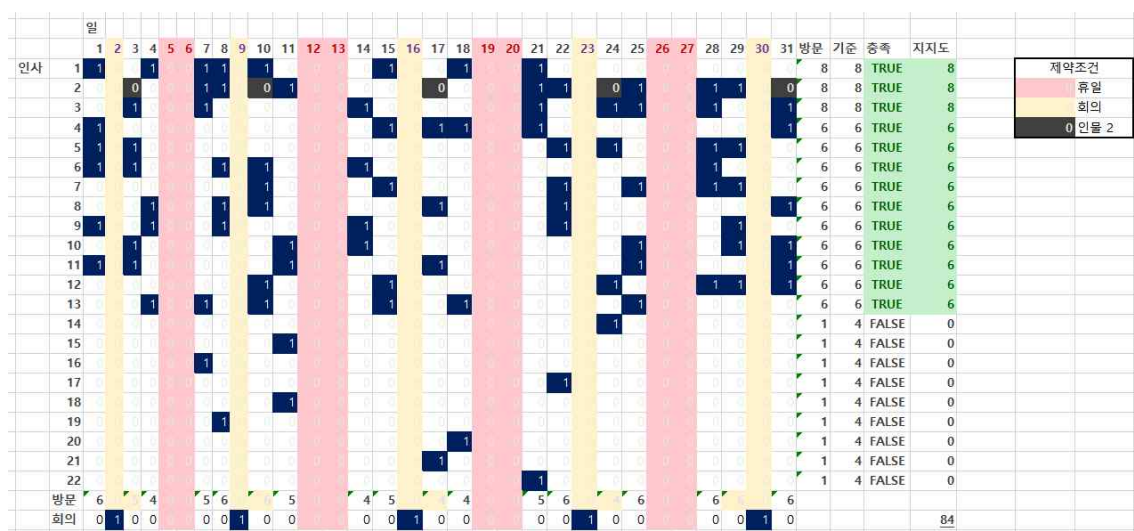
7. 상황변경

1) 실제로 회의는 미리 결정되어 있는 경우가 많으며, 따라서 매주 수요일(2, 9, 16, 23, 30일)에 회의를 갖기로 결정하였습니다. 그리하여 정한 스케줄은 아래와 같습니다.

(1) $M_j = 1$ ($j = 2, 9, 16, 23, 30$) 추가.

$$\text{meetingdays} = \{1, 15, 29\}$$

```
constr9=m.addConstrs(d[j] == 1 for j in meetingdays)
```



2) 10월 9일 한글날 행사, 10월 23일 후보 토론회로 인해 회의 일정을 불가피하게 변경하게 되었습니다. 단, 회의는 같은 주(Week) 안에서만 옮길 수 있다고 가정하였습니다. 또한 1~3주차의 일정은 이미 약속이 되어있어 새롭게 회의가 잡힌 날을 제외하고는 일정 변경이 불가능하며, 4주차, 5주차의 일정은 변경하는 것이 가능하다고 가정하였습니다.

(1) 기존 스케줄 입력

(기존 Optimal Schedule 계산 후)

#결과 입력받기

X=[]

B=[]

for v in m.getVars():

if(i<22):

S.append(v.x)

elif(i<53):

D.append(v.x)

elif(i<735):

X.append(v.x)

else:

B.append(v.x)

i=i+1

#데이터프레임으로 저장

df=pd.DataFrame()

iter=0

for iter in range(22):

quiz=[]

for j in range(31*iter,31*(iter+1)):

quiz.append(X[j])

df=df.append([quiz])

df=df.reset_index(drop=True)

df=df.append([D])

df=df.reset_index(drop=True)

(2) 1, 3주 일정 고정

$$X_{ij} = X_{ij}^{\text{기준}} \quad \text{for all } i, j \in \text{Week1, 3}$$

constr01=m.addConstrs(x[i,j] == df[j][i] for i in range(len(candidate)) for j in week1)

constr02=m.addConstrs(x[i,j] == df[j][i] for i in range(len(candidate)) for j in week3)

(3) 2주차, 4주차의 회의 변경 Constraints

$$X_{ij} = 0 \quad \text{for } (i, j | j = 5, 6, 12, 13, 19, 20, 26, 27, \underline{8, 22})$$

$$M_j = 0 \quad \text{for } (i, j | j = 5, 6, 12, 13, 19, 20, 26, 27, \underline{8, 22})$$

#Weekend에 10월 9일(한글날), 10월 23일(토론회) 추가

weekend={4,5,11,12,18,19,25,26,8,22}

#1, 3, 5주차 회의 고정

$$M_j = 1 \quad \text{for } (j = 1, 15, 29)$$

meetingdays={1,15,29}

constr9=m.addConstrs (d[j] == 1 for j in meetingdays)

(4) 2주차의 변경된 회의일을 제외하고는 나머지 방문 일정은 고정 Constraints

$$X_{ij} = X_{ij}^{\text{기준}} \times (1 - M_j) \quad \text{for all } i, j \in \text{Week2}$$

constr03=m.addConstrs(x[i,j] == df[j][i]*(1-d[j]) for i in range(len(candidate)) for j in week2)

- 기존의 Optimal Schedule

