

Optimization Project

: 롯데리아 Resource scheduling

최적화모형(가) 20192193 김수빈

Contents

1

최적화 대상 소개

2

활용 데이터 수집 방안

3

최적화 문제 정의 및 모델링

4

Mosel 코드 구현 및 풀이

5

최적화 결과 분석

1. 최적화 대상 소개

오지혜	휴무	휴무	B	B	B	C	C
김수영	휴무	B	휴무	휴무	A	A	A
김정민	휴무	A	A	A	휴무	B	B
	03월 22일	03월 23일	03월 24일	03월 25일	03월 26일	03월 27일	03월 28일
	월	화	수	목	금	토	일
	백화점 휴점	물다운		물다운			물다운
1	이효신	休	11 - 16	11 - 16	11 - 16	16 - 21:30	16 - 21:30
2	김수빈		16 - 21:30	16 - 21:30			16:30 - 22
3	구예빈			休	16 - 21:30	16:30 - 21:30	16:30 - 21:30
4	한상미	休	16 - 22			99	11:30-17
5	김가연			16:30 - 22	11 - 16		
6	맹승호		16 - 21:30			16 - 21:30	

롯데리아는 매주 새로 짜여진 스케줄표가 나오는 시스템입니다.

(휴무일이나 운영시간이 매주 달라짐)

그 과정에서 항상 많은 시간이 걸렸고, 오류가 있는 스케줄표로 인한 문제가 생긴 상황도 있어 **최적화 대상**으로 알맞다고 생각했습니다.

2. 활용 데이터 수집 방안

활용 데이터는 지금까지 만들어진 스케줄표를 참고하였습니다.

매주 달라지는 시간대 별로 필요한 인원수를 INPUT 데이터로 하여 하루동안 필요한 알바생의 최소 인원을 목표로 최적화 하였습니다.

3. 최적화 문제 정의 및 모델링

INPUT 데이터: 오른쪽 표와 같이 1시간 단위 별로 필요한 인력 수가 변합니다.

알바생은 하루에 4시간을 연속으로 근무합니다.

TIMES	필요한 최소 인원수
0 (10~11)	1
1 (11~12)	2
2 (12~13)	3
3 (13~14)	3
4 (14~15)	4
5 (15~16)	4
6 (16~17)	3
7 (17~18)	4
8 (18~19)	4
9 (19~20)	3
10 (20~21)	3
11 (21~12)	2

3. 최적화 문제 정의 및 모델링

모델링해보았습니다.

Data

NT : time interval의 수

$t \in T = \{0, \dots, NT-1\}$: time interval의 집합

REQ_t : t 시간대에 필요한 최소 알바 인원 수

Decision Variables

x_t : t 시간대에 근무를 시작하는 인원 수

$t = 4$ 일때 근무 중인 인원 수는
 $x_4 + x_3 + x_2 + x_1$ 과 같다.

Constraints

$x_t + x_{(t-1)} + x_{(t-2)} + x_{(t-3)} \geq REQ_t / t \in \{3, \dots, NT-1\}$

하지만, 롯데리아 매장은 24시간이 아니기 때문에
 t 가 0,1,2 일때 다른 제약식이 필요하다.

Objective Function

$\min \sum_{t \in T} x_t$

근무를 시작하는 인원의 최소합
= 하루 최소 근무 인원

3. 최적화 문제 정의 및 모델링

최종 모델

$$\begin{aligned} & \min \sum_{t \in T} X_t \\ \text{s.t.} \quad & X_t \geq \text{REQ}_t, \quad t = 0 \\ & X_t + X_{(t-1)} \geq \text{REQ}_t, \quad t = 1 \\ & X_t + X_{(t-1)} + X_{(t-2)} \geq \text{REQ}_t, \quad t = 2 \\ & X_t + X_{(t-1)} + X_{(t-2)} + X_{(t-3)} \geq \text{REQ}_t, \quad t \in \{3, \dots, NT-1\} \\ & X_t \geq 0, \quad \forall t \in T \end{aligned}$$

4. Mosel 코드 구현 및 풀이

```
model Scheduling
options noimplicit
uses "mmxprs"

declarations
  NT = 12
  TIMES = 0..NT-1

  REQ : array(TIMES) of integer
  start : array(TIMES) of mpvar
  reqConstr : array(TIMES) of linctr
  minTotal : linctr
end-declarations

initialisations from "input.txt"
  REQ
end-initialisations

minTotal := sum(t in TIMES) start(t)

forall(t in TIMES)
  if t=0 then
    reqConstr(t) := start(t) >= REQ(t)
  elif t=1 then
    reqConstr(t) := start(t) + start(t-1) >= REQ(t)
  elif t=2 then
    reqConstr(t) := start(t) + start(t-1) + start(t-2) >= REQ(t)
  else
    reqConstr(t) := start(t) + start(t-1) + start(t-2) + start(t-3) >= REQ(t)
  end-if

minimise(minTotal)

writeln("하루 최소 근무 인원 : ",getobjval)

forall(t in TIMES)
  writeln(t,"시간에 근무를 시작하는 알바생:",getsol(start(t)))
end-model
```

- Mosel 코드

1	REQ : [1 2 3 3 4 4 3 4 4 3 3 2]
---	-----------------------------------

- INPUT 데이터

- input.txt 파일에 필요한 최소 인원수가 들어가 있어 REQ로 데이터를 받았습니다.
- Start(t)는 t시간에 근무를 시작하는 인원수입니다.
- 이 매장은 24시간 매장이 아니기때문에 오픈시간(t=0~2)에는 근무를 새로 시작하는 인원만 있습니다.

5. 최적화 결과 분석

```
하루 최소 근무 인원 : 9
0시간에 근무를 시작하는 알바생:1
1시간에 근무를 시작하는 알바생:1
2시간에 근무를 시작하는 알바생:1
3시간에 근무를 시작하는 알바생:0
4시간에 근무를 시작하는 알바생:2
5시간에 근무를 시작하는 알바생:1
6시간에 근무를 시작하는 알바생:0
7시간에 근무를 시작하는 알바생:1
8시간에 근무를 시작하는 알바생:2
9시간에 근무를 시작하는 알바생:0
10시간에 근무를 시작하는 알바생:0
11시간에 근무를 시작하는 알바생:0
```

```
Process exited with code: 0
```

- Minimise를 통해 하루에 필요한 최소 알바생 수를 알 수 있게 되었고,
- 알바생들이 언제 근무를 시작해야 하는지도 알 수 있게 되었다.

5. 최적화 결과 분석

```
1 REQ : [ 1 2 3 3 4 4 3 4 4 3 3 2 ]
```

```
하루 최소 근무 인원 : 9
0시간에 근무를 시작하는 알바생:1
1시간에 근무를 시작하는 알바생:1
2시간에 근무를 시작하는 알바생:1
3시간에 근무를 시작하는 알바생:0
4시간에 근무를 시작하는 알바생:2
5시간에 근무를 시작하는 알바생:1
6시간에 근무를 시작하는 알바생:0
7시간에 근무를 시작하는 알바생:1
8시간에 근무를 시작하는 알바생:2
9시간에 근무를 시작하는 알바생:0
10시간에 근무를 시작하는 알바생:0
11시간에 근무를 시작하는 알바생:0
```

```
Process exited with code: 0
```

이제는 하루에 필요한 근무 인원만
정리하여

INPUT 데이터만 수정하면

하루 최소 근무 인원, 각 알바생이
근무에 투입되는 시간이 정리되어
훨씬 빠르고 합리적으로 스케줄표
를 짤 수 있게 되었다.

감사합니다.