



**Amirkabir University of Technology
(Tehran Polytechnic)**

Job Scheduling Problem

Professor:

Dr. F. Hooshmand Khaligh

Teaching Assistants:

Ms. Vista Farahi ,Ms. Fatemeh Vahdat

Student:

Mohammadreza Ardestani (9513004, ardestani.zr@gmail.com)

23, Dec, 2021

0) **Introduction**

0.0) Objective of our problem

0.1) How to Setup and run the code

0.2) Output format

Phase 1) **Formal Model and its Gams Code**

Phase 2) **Python Code**

0.0) Objective of our problem

We have already solved Job Scheduling problem in our Linear Programming course. Now, we would like to rewrite constraints with a close formula.

مسئله زمان بندی کار

شرکت پست به تعدادی کارکنان تمام وقت در روزهای مختلف هفته به شرح زیر نیاز دارد:

روز	شنبه (روز ۱)	یکشنبه (روز ۲)	دوشنبه (روز ۳)	سه شنبه (روز ۴)	چهارشنبه (روز ۵)	پنجشنبه (روز ۶)	جمعه (روز ۷)
تعداد کارکنان تمام وقت مورد نیاز	۱۷	۱۳	۱۵	۱۹	۱۴	۱۶	۱۱

طبق قوانین هر کارمند تمام وقت باید **۵ روز متوالی** کار کند و آنگاه دو روز در مرخصی باشد. مثلاً کارمندی که روزهای دوشنبه تا جمعه کار می کند، شنبه و یکشنبه در مرخصی است. یک مدل ارائه کنید که شرکت بتواند نیاز روزانه اش را با کمترین تعداد کارمند استخدامی تأمین کند.

x_i : تعداد کارمندانی که در روز i ام کار خود را شروع می کنند ($i = 1, 2, \dots, 7$)

مسئله زمان بندی کار

x_i : تعداد کارمندانی که در روز i ام کار خود را شروع می کنند ($i = 1, 2, \dots, 7$)

روز	شنبه (روز ۱)	یکشنبه (روز ۲)	دوشنبه (روز ۳)	سه شنبه (روز ۴)	چهارشنبه (روز ۵)	پنجشنبه (روز ۶)	جمعه (روز ۷)
تعداد کارکنان تمام وقت مورد نیاز	۱۷	۱۳	۱۵	۱۹	۱۴	۱۶	۱۱

$$\min z = x_1 + x_2 + \dots + x_7$$

s. t.

$$x_1 + x_7 + x_6 + x_5 + x_4 \geq 17$$

$$x_2 + x_1 + x_7 + x_6 + x_5 \geq 13$$

$$x_3 + x_2 + x_1 + x_7 + x_6 \geq 15$$

$$x_4 + x_3 + x_2 + x_1 + x_7 \geq 19$$

$$x_5 + x_4 + x_3 + x_2 + x_1 \geq 14$$

$$x_6 + x_5 + x_4 + x_3 + x_2 \geq 16$$

$$x_7 + x_6 + x_5 + x_4 + x_3 \geq 11$$

$$x_i \geq 0, \text{ Integer}$$

Replace these

with close formula

0.1) How to Setup and run the code

Overall, you need to change addresses and have Gams API installed on your PC.

job_scheduling.gms

```
* 1) defining the type of problem
option MIP = CPLEX;
option optcr=0;
option optca=0;
* 2) defining SETs AND PARAMETERS
set
    i;

alias(i,j);
parameter
    staff(i);
* 3) defining variables
$GDXIN C:\\Users\\Mohamadreza.a\\Downloads\\job\\job_data.gdx
$LOAD i, staff
$GDXIN
free Variable z;
integer Variable x(i);
```

```
In [7]: import gams
ws = gams.workspace.GamsWorkspace(working_directory = 'C:\\Users\\Mohamadreza.a\\Downloads\\job\\')
```

```
In [8]: i_python = [str(i) for i in range(1,8)]
staff_python={'1': 17, "2": 13, "3": 15, "4": 19, "5": 14, "6": 16, "7": 11}
```

```
In [9]: db = ws.add_database()

i = db.add_set('i', 1)
for ip in i_python:
    i.add_record(ip)

staff = db.add_parameter_dc('staff', [i])
for ip in i_python:
    staff.add_record(ip).value = staff_python.get(ip)

db.export('C:\\Users\\Mohamadreza.a\\Downloads\\job\\job_data.gdx')
```

```
In [10]: model = ws.add_job_from_file('C:\\Users\\Mohamadreza.a\\Downloads\\job\\job_scheduling.gms')
```

```
In [11]: model.run()
```

```
In [12]: import numpy as np
x = np.zeros(7)
for rec in model.out_db["x"] :
    x[int(rec.key(0)) - 1] = rec.level
print(x)

[6. 4. 1. 8. 0. 4. 0.]
```

0.2) Output format

Since we are using Python to print output, it has an elegant and clear style as follows:

```
In [12]: import numpy as np
x = np.zeros(7)
for rec in model.out_db["x"] :
    x[int(rec.key(0)) - 1] = rec.level
print(x)
```

```
[6. 4. 1. 8. 0. 4. 0.]
```

```
In [ ]: number of people who start their work at day #7
```

number of people who start their work at day #1

Phase 1) Formal Model and Gams Code

For finding off-days for employees based on their start day, one intuitive way is to use Mod operator as follows:

The first off-day is after 5 days and the second off-day is after 6 days.

We can easily find off-days with Mod except those days having off at day #7.

We just simply handle day #7 in another constraint.

$(1+5) \% 7$	$= 6$	✓
$(2+5) \% 7$	$= 0$	✗
$(3+5) \% 7$	$= 1$	✓
$(4+5) \% 7$	$= 2$	✓
$(5+5) \% 7$	$= 3$	✓
$(6+5) \% 7$	$= 4$	✓
$(7+5) \% 7$	$= 5$	✓

```
option optcra=0;
* 2) defining SETs AND PARAMETERS
set
    i; i is ranging from 1 to 7

alias(i,j);
parameter
    staff(i);
* 3) defining variables
$GDXIN C:\Users\Mohamadreza.a\Downloads\job\job_data.gdx
$LOAD i, staff
$GDXIN staff(i): all workers that we need for day #i
free Variable z;
integer Variable x(i); x(i): workers who start at day#i

* 4) introducing equations
equation
    obj, const1,const2;
* 5) defining variables
obj..
    z=e= sum(i,x(i)); <minimizing all workers that we hire>
const1(i)$ (i.val<7)..
    * Here we handle all days excepts day #7
    x(i)+ sum((j)$ ((mod(j.val+5,7)<>i.val) and (mod(j.val+6,7)<>i.val)
        and (j.val<>i.val) ),x(j)) =g= staff(i);
const2(i)$ (i.val=7)..
    * Here we handle day #7
    x(i)+ sum((j)$ ((1<>j.val) and (2<>j.val)
        and (j.val<>i.val) ),x(j)) =g= staff(i);

* 6) introducing the model
model problem1/all/
```

Phase 2) Python Code

```
1: import gams
ws = gams.workspace.GamsWorkspace(working_directory = 'C:\\Users\\Mohamadreza.a\\Downloads\\job')
```

initializing step

```
1: i_python = [str(i) for i in range(1,8)]
staff_python={"1": 17, "2": 13, "3": 15, "4": 19, "5": 14, "6": 16, "7": 11}
```

input data (will be sent to Gams)

```
1: db = ws.add_database()

i = db.add_set('i', 1)
for ip in i_python:
    i.add_record(ip)

staff = db.add_parameter_dc('staff', [i])
for ip in i_python:
    staff.add_record(ip).value = staff_python.get(ip)

db.export('C:\\Users\\Mohamadreza.a\\Downloads\\job\\job_data.gdx')
```

Creating.gdx file and setting up db

```
1: model = ws.add_job_from_file("C:\\Users\\Mohamadreza.a\\Downloads\\job\\job_scheduling.gms")
```

pasing address

```
1: model.run()
```

running model

```
1: import numpy as np
x = np.zeros(7)
for rec in model.out_db["x"] :
    x[int(rec.key(0)) - 1] = rec.level
print(x)

[6. 4. 1. 8. 0. 4. 0.]
```

showing the result :)

```
1:
```

Thanks for your time and attention.