2. 指派问题的 Matlab、Lingo、Mathematica 及 1stOpt 求解对比

参照知乎贴：https://www.zhihu.com/question/49319704，归纳总结了四款数学软件在运筹学指派问题的求解对比，

2.1 指派问题

有 n 项任务要分配给 n 个人去做，每人只完成一项任务，每个任务也只由一个人完成。已知每人完成每项任务所需的时间效率矩阵 C[i,j]如下。问如何分配任务，使完成所有工作所用的总时间最少？

例如：任务为 1，2，3，4，人员为 1，2，3，4，时间效率矩阵 C[i,j]为：

	任务 1	任务 2	任务 3	任务 4
人员 1	2	15	13	4
人员 2	10	4	14	5
人员 3	9	14	16	13
人员 4	7	8	11	9

2.2 模型建立

引入决策变量 x[i,j]，如果让第 i 个人去完成第 j 项任务，则 x[i,j] = 1，否则 x[i,j] = 0,则指派问题的数学模型为：

$$min\ Z=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij}$$

$$s.t.\ \begin{cases}\sum\limits_{j=1}^{n}x_{ij}=1, & i=1,2,\cdots,n,\\ \sum\limits_{i=1}^{n}x_{ij}=1, & j=1,2,\cdots,n,\\ x_{ij}\in\{0,1\}, & i=1,2,\cdots,n;\ j=1,2,\cdots,n.\end{cases}$$

2.3 Matlab 求解

用 MATLAB 求解线性规划或混合整数规划时，决策变量只能是向量（即一维数组），所以需要把上述数学模型中的双下标常量和决策变量都转化成单下标变量，同时约束也需要进行相应的转换。最终求解代码如下。

Matlab 代码：

```
f=[2,10,9,7,15,4,14,8,13,14,16,11,4,5,13,9];
intcon = 1:16;
Aeq=[
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0;
0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0;
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0;
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1;
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1]
beq = ones(8,1);
lb=zeros(1,16);
ub=ones(1,16);

[Y,fval,exitflag] = intlinprog(f,intcon,[],[],Aeq,beq,lb,ub);
x=reshape(Y,4,4)
fval
```

Matlab 结果输出：

```
ans =
        0       0       0       1
        0       1       0       0
        1       0       0       0
        0       0       1       0
Fval = 28
```

新版 MATLAB 提供了更具可读性的问题描述及解决方法。

新版 Matlab 求解代码：

```
c = [2 15 13 4; 10 4 14 5; 9 14 16 13; 7 8 11 9];
prob = optimproblem('ObjectiveSense', 'min');
x = optimvar('x', 4, 4, 'Type', 'integer', 'LowerBound', 0, 'UpperBound', 1);
prob.Objective = sum(sum(c .* x));
prob.Constraints.cond1 = sum(x, 1) == 1;
prob.Constraints.cond2 = sum(x, 2) == 1;
% prob.showproblem
sol = prob.solve;
sol.x
```

Matlab 结果输出：

```
LP:                    Optimal objective value is 28.000000.

Optimal solution found.

Intlinprog stopped at the root node because the
objective value is within a gap tolerance of the optimal value,
options.AbsoluteGapTolerance = 0 (the default value). The intcon variables are
integer within tolerance, options.IntegerTolerance = 1e-05 (the default value).

ans =
     0     0     0     1
     0     1     0     0
     1     0     0     0
     0     0     1     0
```

2.4 Lingo 求解

用 LINGO 来求解这个指派问题，只需要会建立集合及其属性，其它就基本上是对数学模型的直接翻译，无需复杂的转换。

Lingo 求解代码：

```
MODEL:

DATA:
N = 4;
ENDDATA

SETS:

AGENTS/1..N/;
TASKS/1..N/;
MATRIX(AGENTS,TASKS):COST, X;

ENDSETS

DATA:

COST =
2,15,13,4,
10,4,14,5,
9,14,16,13
7,8,11,9;

ENDDATA

MIN = @SUM(AGENTS(I) :
    @SUM(TASKS(J): COST(I,J)*X(I,J)));
```

```
@FOR(AGENTS(I):
    @SUM(TASKS(J):X(I,J)) = 1);

@FOR(TASKS(J):
    @SUM(AGENTS(I):X(I,J)) = 1);

@FOR(AGENTS(I):
    @FOR(TASKS(J) : @GIN(X(I,J))));

END
```

Lingo 结果输出

```
Global optimal solution found.
Objective value:                    28.00000000000000
Objective bound:                    28.00000000000000
Infeasibilities:                    0.000000000000000
Extended solver steps:                              0
Total solver iterations:                            0
Elapsed runtime seconds:                         0.03

Model Class:                                     PILP

Total variables:                   16
Nonlinear variables:                0
Integer variables:                 16

Total constraints:                  9
Nonlinear constraints:              0

Total nonzeros:                    48
Nonlinear nonzeros:                 0


                  Variable           Value        Reduced Cost
                         N        4.000000000000000        0.000000000000000
              COST( 1, 1)        2.000000000000000        0.000000000000000
              COST( 1, 2)        15.00000000000000        0.000000000000000
              COST( 1, 3)        13.00000000000000        0.000000000000000
              COST( 1, 4)        4.000000000000000        0.000000000000000
              COST( 2, 1)        10.00000000000000        0.000000000000000
              COST( 2, 2)        4.000000000000000        0.000000000000000
              COST( 2, 3)        14.00000000000000        0.000000000000000
              COST( 2, 4)        5.000000000000000        0.000000000000000
              COST( 3, 1)        9.000000000000000        0.000000000000000
              COST( 3, 2)        14.00000000000000        0.000000000000000
              COST( 3, 3)        16.00000000000000        0.000000000000000
              COST( 3, 4)        13.00000000000000        0.000000000000000
              COST( 4, 1)        7.000000000000000        0.000000000000000
```

COST(4, 2)	8.000000000000000	0.000000000000000
COST(4, 3)	11.00000000000000	0.000000000000000
COST(4, 4)	9.000000000000000	0.000000000000000
X(1, 1)	0.000000000000000	2.000000000000000
X(1, 2)	0.000000000000000	15.00000000000000
X(1, 3)	0.000000000000000	13.00000000000000
X(1, 4)	1.000000000000000	4.000000000000000
X(2, 1)	0.000000000000000	10.00000000000000
X(2, 2)	1.000000000000000	4.000000000000000
X(2, 3)	0.000000000000000	14.00000000000000
X(2, 4)	0.000000000000000	5.000000000000000
X(3, 1)	1.000000000000000	9.000000000000000
X(3, 2)	0.000000000000000	14.00000000000000
X(3, 3)	0.000000000000000	16.00000000000000
X(3, 4)	0.000000000000000	13.00000000000000
X(4, 1)	0.000000000000000	7.000000000000000
X(4, 2)	0.000000000000000	8.000000000000000
X(4, 3)	1.000000000000000	11.00000000000000
X(4, 4)	0.000000000000000	9.000000000000000

Row	Slack or Surplus	Dual Price
1	28.00000000000000	-1.000000000000000
2	0.000000000000000	0.000000000000000
3	0.000000000000000	0.000000000000000
4	0.000000000000000	0.000000000000000
5	0.000000000000000	0.000000000000000
6	0.000000000000000	0.000000000000000
7	0.000000000000000	0.000000000000000
8	0.000000000000000	0.000000000000000
9	0.000000000000000	0.000000000000000

2.5 Mathematica 求解

首先进行初始化：

```
mat = Table[Subscript[a, i, j], {i, 1, 4}, {j, 1, 4}];
Cmat = {{2, 15, 13, 4}, {10, 4, 14, 5}, {9, 14, 16, 13}, {7, 8, 11, 9}};
```

之后使用 Minimize 函数：

```
Minimize[{Flatten[mat].Flatten[Cmat],
    Table[Sum[mat[[k, i]], {i, 1, 4}], {k, 1, 4}] == {1, 1, 1, 1},
    Table[Sum[mat[[i, k]], {i, 1, 4}], {k, 1, 4}] == {1, 1, 1, 1}}~
  Join~Flatten@
    Table[0 <= Subscript[a, i, j] <= 1, {i, 1, 4}, {j, 1, 4}],
 Flatten[mat], Integers]
```

Mathematica 结果输出

```
{28, {a1.1 -> 0, a1.2 -> 0, a1.3 -> 0, a1.4 -> 1, a2.1 -> 0, a2.2 -> 1, a2.3 -> 0, a2.4 -> 0,
a3.1 -> 1, a3.2 -> 0, a3.3 -> 0, a3.4 -> 0, a4.1 -> 0, a4.2 -> 0, a4.3 -> 1, a4.4 -> 0}}
```

Mathematica 结果列为矩阵

```
In[67]:= mat /.
              ({28, {a1.1 -> 0, a1.2 -> 0, a1.3 -> 0, a1.4 -> 1,
                a2.1 -> 0, a2.2 -> 1, a2.3 -> 0, a2.4 -> 0,
                a3.1 -> 1, a3.2 -> 0, a3.3 -> 0, a3.4 -> 0,
                a4.1 -> 0, a4.2 -> 0, a4.3 -> 1, a4.4 -> 0}} [[2]] //MatrixForm

Out[67]//MatrixForm
      (0     0     0     1
       0     1     0     0
       1     0     0     0
       0     0     1     0)
```

2.6 1stOpt 求解

代码 2-1

```
Constant n=4, c(n,n)=[2,15,13,4,10,4,14,5,9,14,16,13,7,8,11,9];
IntParameter x(n,n);
Algorithm = LP;
MinFunction Sum(i=1:n)(Sum(j=1:n)(c[i,j]*x[i,j]));
                 For(i=1:n)(Sum(j=1:n)(x[i,j])=1);
                 For(j=1:n)(Sum(i=1:n)(x[i,j])=1);
```

结果输出

Objective Function(Min.): 28	x[2,4]: 0
Best Estimated Parameters:	x[3,1]: 1
x[1,1]: 0	x[3,2]: 0
x[1,2]: 0	x[3,3]: 0
x[1,3]: 0	x[3,4]: 0
x[1,4]: 1	x[4,1]: 0
x[2,1]: 0	x[4,2]: 0
x[2,2]: 1	x[4,3]: 1
x[2,3]: 0	x[4,4]: 0

2.7 小结

对于指派问题，四种软件均可获得相同结果的解。

对不同的版本的 Matlab，提供了两种求解方式，第一种比较繁琐，需要进行一些转换工作才能进行求解，第二种相对简单些，两种方式都需要掌握特定命令语句；

Lingo 求解，需要熟知 Lingo 语言中关于集合及其属性的定义，总体上是对数学模型的直接翻译，无需复杂的转换；

Mathematica 代码简短，但命令高度抽象，生涩难懂；

1stOpt 代码最为直观简洁，与模型公式完全一样。