

# Challenge April 2025

## Case Assignments

A solution with OPL CPLEX by Alex Fleischer  
[afleischer@fr.ibm.com](mailto:afleischer@fr.ibm.com)

CPLEX is available within IBM AI platform watsonx.AI  
And OPL is a high level modeling language for CPLEX

The data can be rewritten into the following .dat file

```
//Qualification Level,Minimum Case Complexity,Maximum Case Complexity
qualifications={
<1,1,2>,
<2,1,2>,
<3,1,2>,
<4,1,3>,
<5,2,3>,
<6,2,4>,
<7,2,4>,
<8,3,4>,
<9,3,5>,
<10,4,5>
};
```

```
//Analyst Name,Analyst Level,Total Amount of Assigned Cases,Total Number of Assigned Cases,Analyst Focus Areas,Maximum Allowable Case Amount,Total Cases Maximum Dollar Amount
analysts=
{
<"Tim Smith",10,35000000,12,"Technology, Research, Construction",50000000,75000000>,
<"Sue Rogers",5,5000000,10,"Technology, Research",1000000,7000000>,
<"Sam Howard",8,21000000,9,"Construction, Research",20000000,20000000>,
}
```

```

<"Jill Ryan",9,14700000,10,"Technology, Research,
Construction",1500000,25000000>,
<"Debbie Smith",6,8000000,14,"Research,
Technology",10000000,10000000>,
<"Debbie Bowers",7,6000000,8,"Technology,
Research",1000000,7500000>,
<"Kevin Jones",4,2800000,8,"Research,
Technology",3000000,3000000>,
<"Roger Howland",2,850000,7,"Construction,
Technology",300000,1000000>
};

```

```

//Case Number,Case Amount,Case Complexity,Case Type
cases=
{
<112,50000,3,Technology>,
<113,200000,1,Technology>,
<114,1500000,4,Construction>,
<115,300000,4,Research>,
};

```

And then we can write the model .mod

```

{string} Types={"Technology","Research","Construction"};

//Qualification Level,Minimum Case Complexity,Maximum Case
Complexity
tuple qualification
{
    key int level;
    int minComplexity;
    int maxComplexity;
}
{qualification} qualifications=...;

//Analyst Name,Analyst Level,Total Amount of Assigned
Cases,Total Number of Assigned Cases,Analyst Focus
Areas,Maximum Allowable Case Amount,Total Cases Maximum
Dollar Amount

```

```

tuple analyst
{
    key string name;
    int level;
    float totalAmountOfAssignedcases;
    int totalNumberOfAssignedCases;
    string focusArea;
    float maxcase;
    float maxtotalcase;
}
{analyst} analysts=...;

```

```

// This analysts has this skill ?
int hasThisSkill[analysts][Types];

```

```

execute
{
    for(var a in analysts)
        for(var s in Types)
            if (-1!=a.focusArea.indexOf(s))
                hasThisSkill[a][s]=1;
}

```

```

//Case Number,Case Amount,Case Complexity,Case Type
tuple case
{
    key int number;
    float amount;
    int complexity;
    string type;
}
{case} cases with type in Types=...;

```

```

dvar boolean x[cases][analysts];
dvar int obj;
minimize obj;
subject to
{
    // the firm prefers to minimize the overqualification when analysts

```

```

// are assigned to the cases below their levels.
//obj==sum(a in analysts,c in cases) (x[c][a]*(-
item(qualifications,<a.level>).minComplexity+c.complexity));

obj==sum(a in analysts,c in cases) (x[c][a]*(-c.complexity+a.level));

// One and only one analyst per case
forall(c in cases) sum(a in analysts) x[c][a]==1;

// One analysts can be at most on one case
forall(a in analysts) sum(c in cases) x[c][a]<=1;

//Case must be assigned to an analyst who has the same focus
area as the case type
forall(a in analysts,c in cases:hasThisSkill[a][c.type]==0) x[c][a]==0;

//Analyst can not work on a case that is higher than their maximum
allowable case level
forall(a in analysts,c in cases:a.maxcase<c.amount) x[c][a]==0;

//Analysts can not work on a new case with an amount higher than
their maximum allowed case amount
forall(a in analysts,c in cases:c.amount>a.maxcase) x[c][a]==0;

//Analysts can not work on a new case if it puts them over their
maximum total cases dollar amount
forall(a in analysts,c in
cases:a.totalAmountOfAssignedcases+c.amount>a.maxtotalcase)
x[c][a]==0;

//Analyst Levels must correspond to Case Complexity: higher level
//analysts can work on lower complexity cases between the allowed
Minimum and Maximum
forall(a in analysts,c in
cases:item(qualifications,<a.level>).maxComplexity<c.complexity)
x[c][a]==0;
forall(a in analysts,c in
cases:item(qualifications,<a.level>).minComplexity>c.complexity)
x[c][a]==0;

```

```
}
```

```
execute display_result
```

```
{
```

```
  for(var c in cases) for(var a in analysts) if (x[c][a]==1)
```

```
    writeln("The case ",c.number," is assigned to ",a.name);
```

```
}
```

```
/*
```

That gives

```
// solution with objective 12
```

```
The case 112 is assigned to Sue Rogers
```

```
The case 113 is assigned to Kevin Jones
```

```
The case 114 is assigned to Jill Ryan
```

```
The case 115 is assigned to Debbie Smith
```

```
*/
```

Which gives

```
The case 112 is assigned to Sue Rogers
```

```
The case 113 is assigned to Kevin Jones
```

```
The case 114 is assigned to Jill Ryan
```

```
The case 115 is assigned to Debbie Smith
```