**The Exercise of Intelligent Control Technology**

**Exercise**

Given 3 classes of tasksand 2 classes of machines. Now 5andare scheduled on 1and 2. A machine only tackles a task at a time. The time taken by different machines for different tasks is showed as follows.

The problem is to minimize the makespan such that 10 tasks are completed.

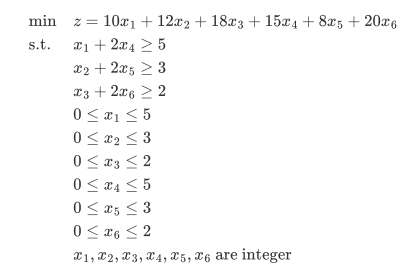
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 10 | 12 | 18 |
|  | 15 | 8 | 20 |

**Solution**

Suppose that are arranged inandin following quantities:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sincecan work at the same time, the model is established as follows:



From the question, we can know that this is a integer programming model, so the solution is achieved through Python’s cplex module. See the appendix code for details. The solution is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 1 | 0 | 0 |
|  | 2 | 2 | 1 |

Sincework at the same time, acompletesin the result, that it can finally complete, which is greater thanrequired by its task. Therefore, after acompletes, another one can only complete, and then immediately start , so the total makespan is reduced by 8 makespan, and the final result is:

So the minimum makespan such that 10 tasks are completed is 68.

**Appendix**

|  |
| --- |
| from docplex.mp.model import Model  # initialize model  cp = Model()  # initialize variables  x1 = cp.integer\_var(ub=5, name="x1")  x2 = cp.integer\_var(ub=3, name="x2")  x3 = cp.integer\_var(ub=2, name="x3")  x4 = cp.integer\_var(ub=5, name="x4")  x5 = cp.integer\_var(ub=3, name="x5")  x6 = cp.integer\_var(ub=2, name="x6")  # defind minimize object function  cp.minimize(10 \* x1 + 12 \* x2 + 18 \* x3 + 15 \* x4 + 8 \* x5 + 20 \* x6)  # defind constraints  cp.add\_constraint(x1 + x2 + x3 + 2 \* (x4 + x5 + x6) >= 10)  cp.add\_constraint(x1 + 2 \* x4 >= 5)  cp.add\_constraint(x2 + 2 \* x5 >= 3)  cp.add\_constraint(x3 + 2 \* x6 >= 2)  # solve the problem  solution = cp.solve()  # print result  if solution:  print(solution)  else:  print("fail") |