

Python读取柔性作业车间调度基准实例mk系列

原创文章 谢师傅season 2020-07-10 20:38:56 846 收藏 11 版权

文章标签： python 算法 计算机应用 柔性作业车间调度 人工蜂群算法 多目标优化

Brandimarte给出了10组柔性作业车间调度实例 (mk01~mk10), 编写代码时, 往往需要将基准实例转换成计算机所能识别的数据结构, 本文提供一种基于Python的实现方法。

基准实例往往如下图所示，需要注意的是，必须严格按照**行号**进行修整，多一个空行，多一个空格数据的读取都可能出错！！

(以下图mk01举例,有效数据仅有11行,如果有12行,则读取失败)

1	10	6	2																																				
2	6	2	1	5	3	4	3	5	3	3	5	2	1	2	3	4	6	2	3	6	5	2	6	1	1	1	3	1	3	6	6	3	6	4	3				
3	5	1	2	6	1	3	1	1	1	2	2	2	6	4	6	3	6	5	2	6	1	1																	
4	5	1	2	6	2	3	4	6	2	3	6	5	2	6	1	1	3	3	4	2	6	6	6	2	1	1	5	5											
5	5	3	6	5	2	6	1	1	1	2	6	1	3	1	3	1	3	5	3	3	5	2	1	2	3	4	6	2											
6	6	3	5	3	3	5	2	1	3	6	5	2	6	1	1	1	2	6	2	1	5	3	4	2	2	6	4	6	3	3	4	2	6	6	6				
7	6	2	3	4	6	2	1	1	2	3	3	4	2	6	6	6	1	2	6	3	6	5	2	6	1	1	2	1	2	1	3	4	2						
8	5	1	6	1	2	1	3	4	2	3	3	4	2	6	6	6	3	2	6	5	1	1	6	1	3	1													
9	5	2	3	4	6	2	3	3	4	2	6	6	6	3	6	5	2	6	1	1	2	6	2	1	2	6	2	6	4	6									
10	6	1	6	1	2	1	1	5	3	6	6	3	6	4	3	1	1	2	3	3	4	2	6	6	6	2	2	6	4	6									
11	6	2	3	4	6	2	3	3	4	2	6	6	6	3	5	3	3	5	2	1	1	6	1	2	2	6	3	4	2	1	1	3	4	2					

数据设置说明

(1) 每一组数据的第一行至少包含2个数字：第一个数字表示工件数，第二个数字表示机器数

(第三个不是必须的,表示每一道工序平均可选择的加工机器数);

(2) 第二行表示**某一个工件**：第一个数字 α_1 表示此工件的总工序数，第二个数字 α_2 表示加工**第一道工序的可选机器数**，接着会有可选机器数的个数的一组数据（机器号、加工时间）；然后是第二道工序的可选机器数，以此类推。

以mk01举例：

10	6	2													
6		2	1	5	3	4	3	5	3	3	5	2	1		
α_1		α_2	α_3		α_4		α_5								

第一行：10个工件，6台可选机器，每道工序平均有2台机器可选择。

第二行: α_1 : 第一个工件有6道工序; α_2 : 第一道工序有2台加工机器可选, 在第1 (α_3) 台机器上的加工时间是5, 在第3 (α_4) 台机器上的加工时间是4; 以此类推, α_5 : 第一道工序有3台加工机器可选.....

实现代码

```

1 class Decode:
2     def __init__(self, inputFile: str, outputFile: str):
3         self.__MAC_INFO = []
4         self.__PRO_INT0 = []
5         self.__proNum = []
6         self.__lines = None
7         self.__input = inputFile
8         self.__output = outputFile
9         self.__listType = {"MAC_INFO": "w", "PRO_INT0": "a"}
10
11     def getMatrix(self):
12         self.__readExample()
13         self.__initMatrix()
14
15         for i in range(len(self.__lines)):
16             lo = 0
17             hi = 0
18             for j in range(self.__proNum[i]):
19                 head = int(self.__lines[i][lo])
20                 hi = lo + 2 * head + 1
21
22                 lo += 1
23                 while lo < hi:
24                     self.__MAC_INFO[i][j].append(int(self.__lines[i][lo]))
25                     self.__PRO_INT0[i][j].append(int(self.__lines[i][lo + 1]))
26
27                 lo += 2
28
29     self.__print("MAC INFO", self.__MAC_INFO)
30     self.__print("PRO INT0", self.__PRO_INT0)

```

 点赞4
 评论5
 分享
 收藏11
 打赏
 举报
 关注
 一键三连

```

30         self.__print("PRO_INT0")
31
32     def __print(self, writePattern):
33         with open(self.__output, self.__listType[writePattern]) as file_object:
34             file_object.write(writePattern + " = [\n")
35             if writePattern == "MAC_INFO":
36                 for i in self.__MAC_INFO:
37                     file_object.write(str(i) + "\n")
38             elif writePattern == "PRO_INT0":
39                 for i in self.__PRO_INT0:
40                     file_object.write(str(i) + "\n")
41             file_object.write("]\n")
42             file_object.write("\n")
43
44     def __initMatrix(self):
45         for i in range(len(self.__proNum)):
46             self.__MAC_INFO.append([])
47             self.__PRO_INT0.append([])
48             for j in range(self.__proNum[i]):
49                 self.__MAC_INFO[i].append([])
50                 self.__PRO_INT0[i].append([])
51
52     def __readExample(self):
53         with open(self.__input) as fileObject:
54             self.__lines = fileObject.readlines()
55
56         # 数据调整
57         del self.__lines[0]
58         for i in range(len(self.__lines)):
59             self.__lines[i] = self.__lines[i].lstrip().rstrip().split(" ")
60             self.__proNum.append(int(self.__lines[i].pop(0)))
61             while "" in self.__lines[i]:
62                 self.__lines[i].remove("")
63
64
65 if __name__ == "__main__":
66     rslt = Decode('Mk01.fjs', "../输出//mk01.txt")
67     rslt.getMatrix()

```

输出结果

```

1  MAC_INFO = [
2  [[1, 3], [5, 3, 2], [3, 6], [6, 2, 1], [3], [6, 3, 4]]
3  [[2], [3], [1], [2, 4], [6, 2, 1]]
4  [[2], [3, 6], [6, 2, 1], [3, 2, 6], [1, 5]]
5  [[6, 2, 1], [2], [3], [5, 3, 2], [3, 6]]
6  [[5, 3, 2], [6, 2, 1], [2], [1, 3], [2, 4], [3, 2, 6]]
7  [[3, 6], [1], [3, 2, 6], [2], [6, 2, 1], [1, 4]]
8  [[6], [1, 4], [3, 2, 6], [2, 5, 1], [3]]
9  [[3, 6], [3, 2, 6], [6, 2, 1], [2], [2, 4]]
10 [[6], [1, 5], [6, 3, 4], [1], [3, 2, 6], [2, 4]]
11 [[3, 6], [3, 2, 6], [5, 3, 2], [6], [2, 4], [1, 4]]
12 ]
13
14 PRO_INT0 = [
15 [[5, 4], [3, 5, 1], [4, 2], [5, 6, 1], [1], [6, 6, 3]]
16 [[6], [1], [2], [6, 6], [5, 6, 1]]
17 [[6], [4, 2], [5, 6, 1], [4, 6, 6], [1, 5]]
18 [[5, 6, 1], [6], [1], [3, 5, 1], [4, 2]]
19 [[3, 5, 1], [5, 6, 1], [6], [5, 4], [6, 6], [4, 6, 6]]
20 [[4, 2], [2], [4, 6, 6], [6], [5, 6, 1], [3, 2]]
21 [[1], [3, 2], [4, 6, 6], [6, 1, 6], [1]]
22 [[4, 2], [4, 6, 6], [5, 6, 1], [6], [6, 6]]
23 [[1], [1, 5], [6, 6, 3], [2], [4, 6, 6], [6, 6]]
24 [[4, 2], [4, 6, 6], [3, 5, 1], [1], [6, 6], [3, 2]]
25 ]

```

以上便完成了车间调度基准测试算例的读取工作

“华为云-东吴杯”2021数字化转型创新应用大赛

广告 关闭

汽车零部件的质量是整车质量的基础，目前人工目视检测的方式不仅存在效率低下，还可能由于人工疲劳造...

粒子群算法 (PSO) 解决柔性作业车间

点赞4

评论5

分享

收藏11

打赏

举报

关注

一键三连