

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# 下面这三行代码是为了画图可以显示中文
from pylab import *
mpl.rcParams['font.sans-serif'] = ['SimHei']
mpl.rcParams['axes.unicode_minus'] = False
```

In [2]:

```
def type_is_same(puck_type, airport_type):
    # 判断飞机的到达 (或起飞) 类型是否与登机口的到达 (或起飞) 类型相同
    airport_type = airport_type.replace(' ', '')
    airport_type = airport_type.split(',')
    #airport_type = [s.split() for s in airport_type]
    #print('puck type is {}'.format(puck_type))
    #print('gate type is {}'.format(airport_type))
    if puck_type in airport_type:
        return True
    else:
        return False

def classify_airport2(all_airports):
    # airport 是所有的登机口
    # classes: 字典用于存储每种类别的登机口
    # classes: 0存储航空楼的登机口, 1存储卫星厅的登机口
    classes = {0:[], 1:[]}

    for airport in all_airports:
        if 'T' in airport['gate']:    # 卫星厅的登机口
            classes[0].append(airport)
        if 'S' in airport['gate']:
            classes[1].append(airport)

    return classes
```

In [3]:

```
def classify_puck2(all_pucks):
    # all_pucks: 列表, 所有转场记录的飞机航班
    # puck_classes: 字典, 每种类别的转场记录飞机航班
    # 此函数将all_pucks进行分类, 并且按照优先级依次放入puck_classes字典中,
    # 字典中的键值越小, 该类别的优先级越高

    puck_classes = {0:[], 1:[], 2:[], 3:[]}
    for puck in all_pucks:
        if puck['a_type'] == 'I':
            if puck['d_type'] == 'D':
                puck_classes[0].append(puck)
            if puck['a_type'] == 'D':
                if puck['d_type'] == 'D':
                    puck_classes[1].append(puck)
            if puck['a_type'] == 'D':
                if puck['d_type'] == 'I':
                    puck_classes[2].append(puck)
            if puck['a_type'] == 'I':
                if puck['d_type'] == 'I':
                    puck_classes[3].append(puck)

    return puck_classes
```

In [4]:

```
def create_gates(gates):
    # puck_data : puck_dataFrame类型, 包含全部登机口的信息
    # puck_data的形状是[num_gates, 6]
    # 返回: airports: 包含全部登机口的列表, 每一个元素是一个登机口
    airports = []
    for i in range(gates.shape[0]):
        gate_data = gates.loc[i, :]
        gate = {'gate':gate_data['登机口'], 'terminal':gate_data['终端厅'], 'region':gate_data['区域'], 'a_type':gate_data['到达类型'],
                'd_type':gate_data['出发类型'], 'body_type':gate_data['机体类别'],
                'puck_records':[], 'assign_flag':False}
        airports.append(gate)

    return airports

def create_pucks(pucks):
    # puck_data : puck_dataFrame类型, 包含全部转场记录的信息
    # puck_data的形状是[num_pucks, 8]
    # 返回: allpucks: 包含全部转场记录的列表, 每一个元素是一个转场记录
    allpucks = []
    for i in range(pucks.shape[0]):
        puck_data = pucks.loc[i, :]
        puck = {'record':puck_data['飞机转场记录号'], 'arrive_time':puck_data['到达相对时间min'], 'a_flight':puck_data['到达航班'], 'a_type':puck_data['到达类型'],
                'plane_type':puck_data['飞机型号'], 'depart_time':puck_data['出发相对时间min'], 'de_flight':puck_data['出发航班'], 'd_type':puck_data['出发类型'],
                'airport':'', 'temporary':0}

        allpucks.append(puck)

    return allpucks
```

In [5]:

```
def plane_type_map(plane_type):
    Wide_body = ['332', '333', '33E', '33H', '33L', '773']
    Narrow_body = ['319', '320', '321', '323', '325', '738', '73A', '73E', '73H', '73L']
    plane_type = str(plane_type)
    if (plane_type in Wide_body):
        return 'W'
    else:
        return 'N'
```

In [6]:

```
def sort_pucks(puck_class):
    # 此函数将同一类别的转场记录按照起飞时间的先后排序
    # puck_class: 列表
    # sort_puckclass: 排序好的转场记录, 按照起飞时间非递减排序
    de_times = [puck['depart_time'] for puck in puck_class]
    sort_index = np.argsort(de_times)
    sort_puckclass = [puck_class[ind] for ind in sort_index]

    return sort_puckclass
```

In [7]:

```

def greedyselector2(sort_puck_class, airport):
    # sort_puck_class: 排序好的转场记录, 列表形式
    # airport: 一个登机口类实例
    start_times = [puck['arrive_time'] for puck in sort_puck_class]
    depart_times = [puck['depart_time'] for puck in sort_puck_class]

    j = 0
    while(sort_puck_class[j]['airport'] != ''):
        j = j + 1

    if airport['assign_flag'] == False: # 登机口没有被分配
        airport['busy_time'] = np.zeros(288)
        sp_ind = max(int(start_times[j]/5)-1, 0)
        ep_ind = int(depart_times[j]/5)
        if sort_puck_class[j]['plane_type'] == airport['body_type']:
            if (type_is_same(sort_puck_class[j]['a_type'], airport['a_type']) &
                (type_is_same(sort_puck_class[j]['d_type'], airport['d_type']
)))):

                print('类型相匹配...')
                sort_puck_class[j]['airport'] = airport['gate']
                airport['puck_records'].append(sort_puck_class[j]['record'])
                airport['busy_time'][sp_ind:ep_ind] = 1
                airport['assign_flag'] = True

        k = j
        for i in range(j+1, len(sort_puck_class)):
            if sort_puck_class[i]['plane_type'] == airport['body_type']: # 飞机
类型与登机口的机体类别相同
                # 飞机到达的类型与出发类型和登机口类型均吻合
                if (type_is_same(sort_puck_class[i]['a_type'], airport['a_type']
)) &
                    (type_is_same(sort_puck_class[i]['d_type'], airport['d_type']
))):

                        if start_times[i]>=depart_times[k]:
                            if sort_puck_class[i]['airport']=='': # 如果该转场记录没
有被分配
                                sort_puck_class[i]['airport'] = airport['gate']
                                k = i
                                airport['puck_records'].append(sort_puck_class[k]['r
ecord'])

                                if start_times[i] == 0:
                                    s_ind = 0
                                else:
                                    s_ind = int(start_times[i]/5)
                                    e_ind = int(depart_times[i]/5)
                                    airport['busy_time'][s_ind:e_ind] = 1

                                airport['assign_flag'] = True

        else:
            for i in range(j+1, len(sort_puck_class)):
                if sort_puck_class[i]['plane_type'] == airport['body_type']: # 飞机
类型与登机口的机体类别相同
                    # 飞机到达的类型与出发类型和登机口类型均吻合
                    if (type_is_same(sort_puck_class[i]['a_type'], airport['a_type']
)) &

```

```

        (type_is_same(sort_puck_class[i]['d_type'], airport['d_type'
])))):
分配
        if sort_puck_class[i]['airport']=='':    # 如果该转场记录没有被

            puck_time = np.zeros(288)
            if start_times[i] == 0:
                s_ind = 0
            else:
                s_ind = int(start_times[i]/5)
            e_ind = int(depart_times[i]/5)
            #print('该记录的起止时间下标分别是{}和{}'.format(s_ind, e_in
d))

            puck_time[s_ind:e_ind] = 1
            temp_time = puck_time + airport['busy_time']
            if np.max(temp_time)<=1:
                print('可以安排插入航班.....')
                airport['busy_time'] = temp_time
                sort_puck_class[i]['airport'] = airport['gate']
                airport['puck_records'].append(sort_puck_class[i]['r
ecord'])

            print('gates{} has assigned {}'.format(airport['gate'],airport['puck_record
s']))
            return sort_puck_class, airport

```

In [8]:

```

def assign_puck2(puck_class, gate_class):
    if len(puck_class)==0 or len(gate_class)==0:
        return puck_class, gate_class
    sort_puck_class = sort_pucks(puck_class)

    for i in range(len(gate_class)):
        puck_not_assign = [puck for puck in sort_puck_class if puck['airport']=='
']
        if len(puck_not_assign)==0:
            break
        sort_puck_class, airport = greedyselector2(sort_puck_class, gate_class[i
])
        gate_class[i] = airport

    return sort_puck_class, gate_class

```

In [9]:

```
gates = pd.read_csv('./gates (1).csv')
new_gates = gates[['登机口', '终端厅', '区域', '到达类型', '出发类型', '机体类别']]

puck_data = pd.read_csv('./puck_data.csv', encoding='gbk')
cols = ['飞机转场记录号', '到达相对时间min', '到达航班', '到达类型',
        '飞机型号', '出发相对时间min', '出发航班', '出发类型']
puck_data = puck_data[cols]

airports = create_gates(new_gates)
allpucks = create_pucks(puck_data)

puck_classes = classify_puck2(allpucks)
gate_classes = classify_airport2(airports)

# single_type_gate = [0, 1, 3, 4, 9, 10, 12, 13]
# multi_type_gate = [2, 5, 6, 7, 8, 11, 14, 15, 16, 17]
# single_gate_classes = [gate_classes[code] for code in single_type_gate]
# multi_gate_classes = [gate_classes[code] for code in multi_type_gate]
```

In [10]:

```
# 用航空楼的登机口尽可能先安排到达类型为国际, 出发类型为国内的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[0], gate_classes[0])
puck_classes[0] = at_puck; gate_classes[0] = at_gate
```

```
gatesT1 has assigned []
gatesT2 has assigned []
gatesT3 has assigned []
gatesT4 has assigned []
类型相匹配...
gatesT5 has assigned ['PK452', 'PK483', 'PK331']
gatesT6 has assigned ['PK374']
类型相匹配...
gatesT7 has assigned ['PK459', 'PK495', 'PK278', 'PK317', 'PK353',
'PK393']
类型相匹配...
gatesT8 has assigned ['PK467', 'PK269', 'PK285', 'PK311', 'PK351',
'PK427']
类型相匹配...
gatesT9 has assigned ['PK444', 'PK270', 'PK290', 'PK329', 'PK359',
'PK404']
gatesT10 has assigned []
gatesT11 has assigned []
gatesT12 has assigned []
gatesT13 has assigned []
gatesT14 has assigned []
gatesT15 has assigned []
gatesT16 has assigned []
gatesT17 has assigned []
gatesT18 has assigned []
gatesT19 has assigned []
gatesT20 has assigned []
gatesT21 has assigned []
类型相匹配...
gatesT22 has assigned ['PK185', 'PK268', 'PK296', 'PK366']
gatesT23 has assigned []
gatesT24 has assigned []
gatesT25 has assigned []
gatesT26 has assigned []
gatesT27 has assigned []
gatesT28 has assigned []
```

In [11]:

```
# 用航空楼的登机口尽可能先安排到达类型和出发类型均为国内的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[1], gate_classes[0])
puck_classes[1] = at_puck; gate_classes[0] = at_gate

gatesT1 has assigned []
gatesT2 has assigned []
gatesT3 has assigned []
gatesT4 has assigned []
gatesT5 has assigned ['PK452', 'PK483', 'PK331']
gatesT6 has assigned ['PK374']
可以安排插入航班.....
gatesT7 has assigned ['PK459', 'PK495', 'PK278', 'PK317', 'PK353',
'PK393', 'PK440']
可以安排插入航班.....
gatesT8 has assigned ['PK467', 'PK269', 'PK285', 'PK311', 'PK351',
'PK427', 'PK396']
gatesT9 has assigned ['PK444', 'PK270', 'PK290', 'PK329', 'PK359',
'PK404']
类型相匹配...
gatesT10 has assigned ['PK179', 'PK476', 'PK259', 'PK276', 'PK293',
'PK325', 'PK355', 'PK395', 'PK429']
类型相匹配...
gatesT11 has assigned ['PK173', 'PK473', 'PK267', 'PK284', 'PK309',
'PK339', 'PK362', 'PK398', 'PK430']
类型相匹配...
gatesT12 has assigned ['PK150', 'PK478', 'PK273', 'PK283', 'PK312',
'PK348', 'PK388', 'PK415', 'PK438']
类型相匹配...
gatesT13 has assigned ['PK188', 'PK477', 'PK266', 'PK288', 'PK323',
'PK360', 'PK406', 'PK431']
类型相匹配...
gatesT14 has assigned ['PK447', 'PK484', 'PK277', 'PK294', 'PK335',
'PK380', 'PK413', 'PK437']
类型相匹配...
gatesT15 has assigned ['PK180', 'PK482', 'PK279', 'PK346', 'PK414',
'PK439']
类型相匹配...
gatesT16 has assigned ['PK175', 'PK479', 'PK305', 'PK345', 'PK416']
类型相匹配...
gatesT17 has assigned ['PK192', 'PK486', 'PK300', 'PK385', 'PK417']
类型相匹配...
gatesT18 has assigned ['PK117', 'PK488', 'PK306', 'PK376', 'PK418']
类型相匹配...
gatesT19 has assigned ['PK443', 'PK490', 'PK299', 'PK364', 'PK419']
类型相匹配...
gatesT20 has assigned ['PK182', 'PK494', 'PK310', 'PK367', 'PK422']
类型相匹配...
gatesT21 has assigned ['PK170', 'PK489', 'PK282', 'PK369', 'PK423']
可以安排插入航班.....
gatesT22 has assigned ['PK185', 'PK268', 'PK296', 'PK366', 'PK424']
gatesT23 has assigned []
gatesT24 has assigned []
gatesT25 has assigned []
gatesT26 has assigned []
gatesT27 has assigned []
gatesT28 has assigned []
```


In [12]:

```
# 用卫星厅的登机口尽可能先安排到达类型为国内, 出发类型为国际的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[2], gate_classes[1])
puck_classes[2] = at_puck; gate_classes[1] = at_gate
```

```
gatesS1 has assigned []
gatesS2 has assigned []
gatesS3 has assigned []
gatesS4 has assigned []
gatesS5 has assigned []
gatesS6 has assigned []
gatesS7 has assigned []
gatesS8 has assigned []
gatesS9 has assigned []
gatesS10 has assigned []
gatesS11 has assigned []
gatesS12 has assigned []
gatesS13 has assigned []
gatesS14 has assigned []
gatesS15 has assigned []
gatesS16 has assigned []
gatesS17 has assigned []
gatesS18 has assigned []
gatesS19 has assigned []
gatesS20 has assigned []
gatesS21 has assigned []
gatesS22 has assigned []
gatesS23 has assigned []
gatesS24 has assigned []
gatesS25 has assigned []
gatesS26 has assigned []
gatesS27 has assigned []
gatesS28 has assigned []
gatesS29 has assigned []
gatesS30 has assigned []
gatesS31 has assigned []
gatesS32 has assigned []
gatesS33 has assigned []
gatesS34 has assigned []
gatesS35 has assigned []
gatesS36 has assigned []
gatesS37 has assigned []
gatesS38 has assigned []
gatesS39 has assigned []
gatesS40 has assigned []
gatesS41 has assigned []
```

In [13]:

```
# 用卫星厅的登机口尽可能先安排到达类型和出发类型均为国际的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[3], gate_classes[1])
puck_classes[3] = at_puck; gate_classes[1] = at_gate

gatesS1 has assigned []
gatesS2 has assigned []
gatesS3 has assigned []
gatesS4 has assigned []
gatesS5 has assigned []
gatesS6 has assigned []
gatesS7 has assigned []
gatesS8 has assigned []
gatesS9 has assigned []
gatesS10 has assigned []
gatesS11 has assigned ['PK470', 'PK260', 'PK287', 'PK324', 'PK358',
'PK409']
gatesS12 has assigned ['PK450', 'PK261', 'PK292', 'PK334', 'PK361',
'PK421']
gatesS13 has assigned ['PK465', 'PK274', 'PK303', 'PK344', 'PK386']
gatesS14 has assigned []
gatesS15 has assigned []
gatesS16 has assigned []
gatesS17 has assigned []
gatesS18 has assigned []
gatesS19 has assigned []
gatesS20 has assigned []
gatesS21 has assigned []
gatesS22 has assigned []
gatesS23 has assigned []
gatesS24 has assigned []
gatesS25 has assigned []
gatesS26 has assigned []
gatesS27 has assigned []
gatesS28 has assigned []
gatesS29 has assigned []
gatesS30 has assigned []
类型相匹配...
gatesS31 has assigned ['PK107', 'PK449', 'PK304', 'PK347']
类型相匹配...
gatesS32 has assigned ['PK129', 'PK454', 'PK316', 'PK411']
类型相匹配...
gatesS33 has assigned ['PK131', 'PK461', 'PK333', 'PK400']
gatesS34 has assigned ['PK464', 'PK308']
gatesS35 has assigned ['PK456', 'PK382']
gatesS36 has assigned ['PK460', 'PK336']
gatesS37 has assigned ['PK468', 'PK338']
gatesS38 has assigned ['PK466', 'PK340']
gatesS39 has assigned ['PK453', 'PK349']
gatesS40 has assigned ['PK463', 'PK363']
gatesS41 has assigned ['PK471', 'PK370']
```

In [14]:

```
# 用卫星厅的登机口尽可能安排到达类型为国际且出发类型为国内, 目前还没有安排的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[0], gate_classes[1])
puck_classes[0] = at_puck; gate_classes[1] = at_gate
```

```
gatesS1 has assigned []
gatesS2 has assigned []
gatesS3 has assigned []
gatesS4 has assigned []
gatesS5 has assigned []
gatesS6 has assigned []
gatesS7 has assigned []
gatesS8 has assigned []
gatesS9 has assigned []
gatesS10 has assigned []
gatesS11 has assigned ['PK470', 'PK260', 'PK287', 'PK324', 'PK358',
'PK409']
gatesS12 has assigned ['PK450', 'PK261', 'PK292', 'PK334', 'PK361',
'PK421']
gatesS13 has assigned ['PK465', 'PK274', 'PK303', 'PK344', 'PK386']
gatesS14 has assigned []
gatesS15 has assigned []
gatesS16 has assigned []
gatesS17 has assigned []
gatesS18 has assigned []
gatesS19 has assigned []
gatesS20 has assigned []
gatesS21 has assigned []
gatesS22 has assigned []
gatesS23 has assigned []
gatesS24 has assigned []
gatesS25 has assigned []
gatesS26 has assigned []
gatesS27 has assigned []
gatesS28 has assigned []
gatesS29 has assigned []
gatesS30 has assigned []
gatesS31 has assigned ['PK107', 'PK449', 'PK304', 'PK347']
gatesS32 has assigned ['PK129', 'PK454', 'PK316', 'PK411']
gatesS33 has assigned ['PK131', 'PK461', 'PK333', 'PK400']
gatesS34 has assigned ['PK464', 'PK308']
gatesS35 has assigned ['PK456', 'PK382']
gatesS36 has assigned ['PK460', 'PK336']
gatesS37 has assigned ['PK468', 'PK338']
gatesS38 has assigned ['PK466', 'PK340']
gatesS39 has assigned ['PK453', 'PK349']
gatesS40 has assigned ['PK463', 'PK363']
gatesS41 has assigned ['PK471', 'PK370']
```

In [15]:

```
# 用卫星厅的登机口尽可能安排到达类型和出发类型均为国内, 目前还没有安排的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[1], gate_classes[1])
puck_classes[1] = at_puck; gate_classes[1] = at_gate
```

类型相匹配...

gatesS1 has assigned ['PK446', 'PK501', 'PK298', 'PK356', 'PK425']

类型相匹配...

gatesS2 has assigned ['PK136', 'PK498', 'PK307', 'PK381', 'PK433']

类型相匹配...

gatesS3 has assigned ['PK147', 'PK496', 'PK315', 'PK383', 'PK435']

类型相匹配...

gatesS4 has assigned ['PK187', 'PK493', 'PK297', 'PK401']

类型相匹配...

gatesS5 has assigned ['PK442', 'PK497', 'PK291', 'PK399']

类型相匹配...

gatesS6 has assigned ['PK171', 'PK491', 'PK320', 'PK407']

类型相匹配...

gatesS7 has assigned ['PK181', 'PK500', 'PK326', 'PK403']

类型相匹配...

gatesS8 has assigned ['PK155', 'PK254', 'PK357']

类型相匹配...

gatesS9 has assigned ['PK191', 'PK264', 'PK408']

类型相匹配...

gatesS10 has assigned ['PK195', 'PK256', 'PK410']

gatesS11 has assigned ['PK470', 'PK260', 'PK287', 'PK324', 'PK358',
'PK409']

gatesS12 has assigned ['PK450', 'PK261', 'PK292', 'PK334', 'PK361',
'PK421']

gatesS13 has assigned ['PK465', 'PK274', 'PK303', 'PK344', 'PK386']

类型相匹配...

gatesS14 has assigned ['PK184', 'PK272', 'PK412']

类型相匹配...

gatesS15 has assigned ['PK448', 'PK387']

类型相匹配...

gatesS16 has assigned ['PK193', 'PK342']

类型相匹配...

gatesS17 has assigned ['PK168', 'PK389']

类型相匹配...

gatesS18 has assigned ['PK197', 'PK390']

类型相匹配...

gatesS19 has assigned ['PK174', 'PK392']

类型相匹配...

gatesS20 has assigned ['PK194', 'PK394']

类型相匹配...

gatesS21 has assigned ['PK102', 'PK428']

类型相匹配...

gatesS22 has assigned ['PK145']

类型相匹配...

gatesS23 has assigned ['PK196']

类型相匹配...

gatesS24 has assigned ['PK441']

类型相匹配...

gatesS25 has assigned ['PK062']

类型相匹配...

gatesS26 has assigned ['PK072']

类型相匹配...

gatesS27 has assigned ['PK480']

In [16]:

```
# 用航空楼的登机口尽可能安排到达类型为国内且出发类型为国际，目前还没有安排的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[2], gate_classes[0])
puck_classes[2] = at_puck; gate_classes[0] = at_gate
```

```
gatesT1 has assigned []
gatesT2 has assigned []
gatesT3 has assigned []
gatesT4 has assigned []
gatesT5 has assigned ['PK452', 'PK483', 'PK331']
可以安排插入航班.....
可以安排插入航班.....
gatesT6 has assigned ['PK374', 'PK487', 'PK280']
gatesT7 has assigned ['PK459', 'PK495', 'PK278', 'PK317', 'PK353',
'PK393', 'PK440']
gatesT8 has assigned ['PK467', 'PK269', 'PK285', 'PK311', 'PK351',
'PK427', 'PK396']
gatesT9 has assigned ['PK444', 'PK270', 'PK290', 'PK329', 'PK359',
'PK404']
gatesT10 has assigned ['PK179', 'PK476', 'PK259', 'PK276', 'PK293',
'PK325', 'PK355', 'PK395', 'PK429']
gatesT11 has assigned ['PK173', 'PK473', 'PK267', 'PK284', 'PK309',
'PK339', 'PK362', 'PK398', 'PK430']
gatesT12 has assigned ['PK150', 'PK478', 'PK273', 'PK283', 'PK312',
'PK348', 'PK388', 'PK415', 'PK438']
gatesT13 has assigned ['PK188', 'PK477', 'PK266', 'PK288', 'PK323',
'PK360', 'PK406', 'PK431']
gatesT14 has assigned ['PK447', 'PK484', 'PK277', 'PK294', 'PK335',
'PK380', 'PK413', 'PK437']
gatesT15 has assigned ['PK180', 'PK482', 'PK279', 'PK346', 'PK414',
'PK439']
gatesT16 has assigned ['PK175', 'PK479', 'PK305', 'PK345', 'PK416']
gatesT17 has assigned ['PK192', 'PK486', 'PK300', 'PK385', 'PK417']
gatesT18 has assigned ['PK117', 'PK488', 'PK306', 'PK376', 'PK418']
gatesT19 has assigned ['PK443', 'PK490', 'PK299', 'PK364', 'PK419']
可以安排插入航班.....
可以安排插入航班.....
gatesT20 has assigned ['PK182', 'PK494', 'PK310', 'PK367', 'PK422',
'PK281', 'PK341']
gatesT21 has assigned ['PK170', 'PK489', 'PK282', 'PK369', 'PK423']
可以安排插入航班.....
gatesT22 has assigned ['PK185', 'PK268', 'PK296', 'PK366', 'PK424',
'PK332']
类型相匹配...
gatesT23 has assigned ['PK106', 'PK352']
gatesT24 has assigned []
gatesT25 has assigned []
gatesT26 has assigned []
gatesT27 has assigned []
gatesT28 has assigned []
```

In [17]:

```
# 用航空楼的登机口尽可能安排到达类型和出发类型均为国际, 目前还没有安排的转场飞机
at_puck, at_gate = assign_puck2(puck_classes[3], gate_classes[0])
puck_classes[3] = at_puck; gate_classes[0] = at_gate
```

类型相匹配...

gatesT1 has assigned ['PK149', 'PK455', 'PK314', 'PK391']

类型相匹配...

gatesT2 has assigned ['PK104', 'PK328']

类型相匹配...

gatesT3 has assigned ['PK108', 'PK377']

gatesT4 has assigned ['PK373']

gatesT5 has assigned ['PK452', 'PK483', 'PK331']

gatesT6 has assigned ['PK374', 'PK487', 'PK280']

gatesT7 has assigned ['PK459', 'PK495', 'PK278', 'PK317', 'PK353', 'PK393', 'PK440']

gatesT8 has assigned ['PK467', 'PK269', 'PK285', 'PK311', 'PK351', 'PK427', 'PK396']

gatesT9 has assigned ['PK444', 'PK270', 'PK290', 'PK329', 'PK359', 'PK404']

gatesT10 has assigned ['PK179', 'PK476', 'PK259', 'PK276', 'PK293', 'PK325', 'PK355', 'PK395', 'PK429']

gatesT11 has assigned ['PK173', 'PK473', 'PK267', 'PK284', 'PK309', 'PK339', 'PK362', 'PK398', 'PK430']

gatesT12 has assigned ['PK150', 'PK478', 'PK273', 'PK283', 'PK312', 'PK348', 'PK388', 'PK415', 'PK438']

gatesT13 has assigned ['PK188', 'PK477', 'PK266', 'PK288', 'PK323', 'PK360', 'PK406', 'PK431']

gatesT14 has assigned ['PK447', 'PK484', 'PK277', 'PK294', 'PK335', 'PK380', 'PK413', 'PK437']

gatesT15 has assigned ['PK180', 'PK482', 'PK279', 'PK346', 'PK414', 'PK439']

gatesT16 has assigned ['PK175', 'PK479', 'PK305', 'PK345', 'PK416']

gatesT17 has assigned ['PK192', 'PK486', 'PK300', 'PK385', 'PK417']

gatesT18 has assigned ['PK117', 'PK488', 'PK306', 'PK376', 'PK418']

gatesT19 has assigned ['PK443', 'PK490', 'PK299', 'PK364', 'PK419']

gatesT20 has assigned ['PK182', 'PK494', 'PK310', 'PK367', 'PK422', 'PK281', 'PK341']

gatesT21 has assigned ['PK170', 'PK489', 'PK282', 'PK369', 'PK423']

gatesT22 has assigned ['PK185', 'PK268', 'PK296', 'PK366', 'PK424', 'PK332']

gatesT23 has assigned ['PK106', 'PK352']

gatesT24 has assigned ['PK457']

gatesT25 has assigned ['PK458']

gatesT26 has assigned []

gatesT27 has assigned []

gatesT28 has assigned []

In [18]:

```
gate_sum = 0
puck_sum = 0

assign_pucks = []; assign_gates = []
for clas in gate_classes.keys():
    for gate in gate_classes[clas]:
        if gate['assign_flag']==True:
            assign_pucks.append(gate['puck_records'])
            assign_gates.append(gate['gate'])
            gate_sum += 1
            puck_sum += len(gate['puck_records'])
```

In [19]:

```
gate_sum
puck_sum
```

Out[19]:

63

Out[19]:

234

In [20]:

```
assign_dict = dict(zip(assign_gates, assign_pucks))
```

In [21]:

```
import csv
# 从字典写入csv文件

csvFile3 = open('问题二答案.csv', 'w', newline='')
writer2 = csv.writer(csvFile3)
for key in assign_dict:
    writer2.writerow([key, assign_dict[key]])
csvFile3.close()
```


Out[21]:

43

Out[21]:

25

Out[21]:

25

Out[21]:

14

Out[21]:

34

Out[21]:

34

Out[21]:

70

Out[21]:

70

Out[21]:

61

Out[21]:

89

Out[21]:

89

Out[21]:

89

Out[21]:

80

Out[21]:

80

Out[21]:

62

Out[21]:

53

Out[21]:

53

Out[21]:

53

Out[21]:

53

Out[21]:

71

Out[21]:

53

Out[21]:

62

Out[21]:

26

Out[21]:

15

Out[21]:

15

Out[21]:

52

Out[21]:

52

Out[21]:

52

Out[21]:

43

Out[21]:

43

Out[21]:

43

Out[21]:

43

Out[21]:

34

Out[21]:

34

Out[21]:

35

Out[21]:

62

Out[21]:

62

Out[21]:

53

Out[21]:

35

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

15

Out[21]:

15

Out[21]:

15

Out[21]:

15

Out[21]:

15

Out[21]:

15

Out[21]:

44

Out[21]:

44

Out[21]:

44

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

Out[21]:

26

画图

In [22]:

```
num_assign_pucks = [len(pucks) for pucks in assign_pucks]
assign_dict = dict(zip(assign_gates, assign_pucks))
assign_dict1 = dict(zip(assign_gates, num_assign_pucks))
assigns = pd.DataFrame(assign_dict1, index=[0])
assigns = assigns.T
```

画出被使用的登机口安排的航班数量图

In [23]:

```
plt.figure(figsize=(20, 10))
x = list(assigns.index)
plt.bar(x, assigns[0]*2, facecolor='b')
plt.xlabel('登机口', fontsize=18)
plt.ylabel('登机口分配的总航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('登机口航班分配情况', fontsize=18)
plt.show()
```

Out[23]:

<Figure size 1440x720 with 0 Axes>

Out[23]:

<BarContainer object of 63 artists>

Out[23]:

Text(0.5,0,'登机口')

Out[23]:

Text(0,0.5,'登机口分配的总航班数量')

Out[23]:

```
([0,  
 1,  
 2,  
 3,  
 4,  
 5,  
 6,  
 7,  
 8,  
 9,  
10,  
11,  
12,  
13,  
14,  
15,  
16,  
17,  
18,  
19,  
20,  
21,  
22,  
23,  
24,  
25,  
26,  
27,  
28,  
29,  
30,  
31,  
32,  
33,  
34,  
35,  
36,  
37,  
38,  
39,  
40,  
41,  
42,  
43,  
44,  
45,  
46,  
47,  
48,  
49,  
50,  
51,  
52,  
53,  
54,  
55,  
56,  
57,  
58,
```

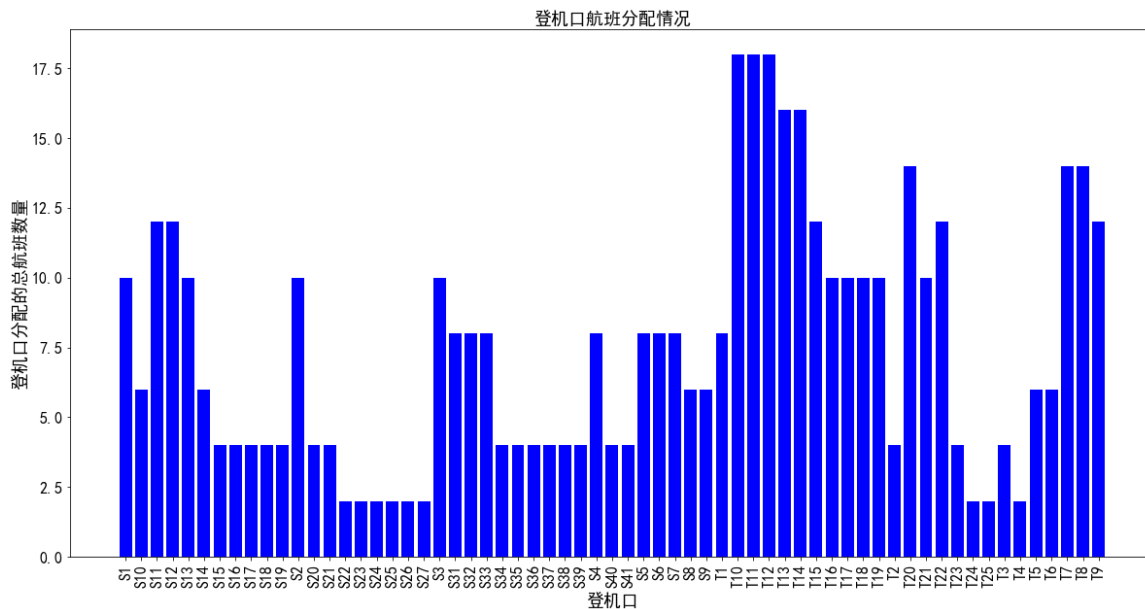
```
59,
60,
61,
62],
<a list of 63 Text xticklabel objects>)
```

Out[23]:

```
(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. ]),
<a list of 9 Text yticklabel objects>)
```

Out[23]:

Text(0.5,1,'登机口航班分配情况')



按照宽体机和窄体机画出登机口安排的航班数量

In [24]:

```
wide_gates = [airport['gate'] for airport in airports if airport['body_type']=='W']
narrow_gates = [airport['gate'] for airport in airports if airport['body_type']=='N']
```

In [25]:

```
narrow_assign_num = {}; wide_assign_num = {}
for gate in assign_dict.keys():
    if gate in wide_gates:
        #print(len(assign_dict[gate]))
        wide_assign_num[gate] = len(assign_dict[gate])
    else:
        #print('narrow'+str(len(assign_dict[gate])))
        narrow_assign_num[gate] = len(assign_dict[gate])
```

In [26]:

```
narrow_assign_num = pd.DataFrame(narrow_assign_num, index=[0]).T
wide_assign_num = pd.DataFrame(wide_assign_num, index=[0]).T
```


In [27]:

```
plt.figure(figsize=(20, 10))
x = list(narrow_assign_num.index)
plt.bar(x, narrow_assign_num[0]*2, facecolor='b')
plt.xlabel('窄体机登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('窄体登机口航班分配情况', fontsize=18)
plt.show()
```

Out[27]:

<Figure size 1440x720 with 0 Axes>

Out[27]:

<BarContainer object of 44 artists>

Out[27]:

Text(0.5,0,'窄体机登机口')

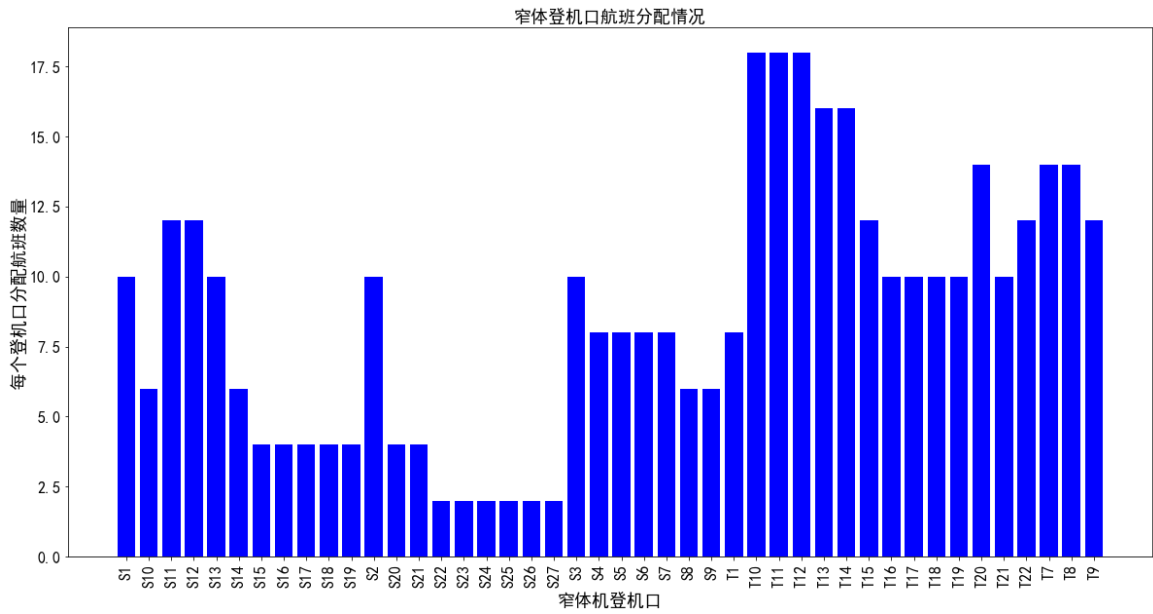
Out[27]:

Text(0,0.5,'每个登机口分配航班数量')

Out[27]:

```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37,
38,
39,
40,
41,
42,
43],
<a list of 44 Text xticklabel objects>)
```

```
Out[27]:  
  
(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. ]),  
 <a list of 9 Text yticklabel objects>)  
  
Out[27]:  
  
Text(0.5,1,'窄体登机口航班分配情况')
```



In [28]:

```
plt.figure(figsize=(20, 10))
x = list(wide_assign_num.index)
plt.bar(x, wide_assign_num[0]*2, facecolor='b')
plt.xlabel('宽体机登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=14)
plt.yticks(fontsize=14)
plt.title('宽体机登机口航班分配情况', fontsize=18)
plt.show()
```

Out[28]:

<Figure size 1440x720 with 0 Axes>

Out[28]:

<BarContainer object of 19 artists>

Out[28]:

Text(0.5,0,'宽体机登机口')

Out[28]:

Text(0,0.5,'每个登机口分配航班数量')

Out[28]:

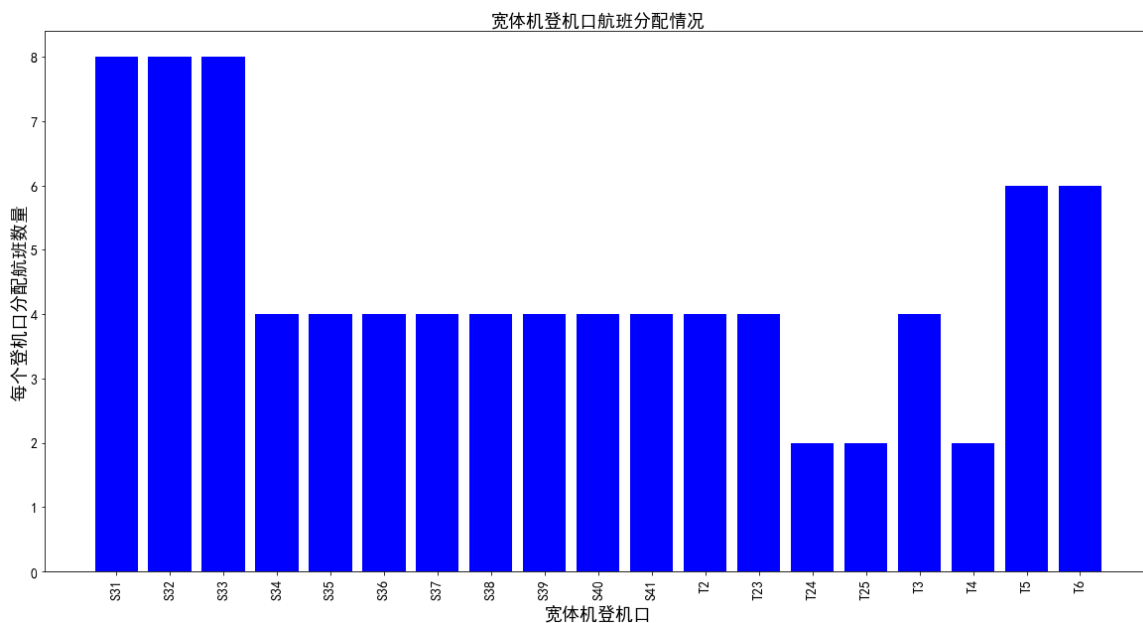
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18],
<a list of 19 Text xticklabel objects>)

Out[28]:

(array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.]),
<a list of 10 Text yticklabel objects>)

Out[28]:

Text(0.5,1,'宽体机登机口航班分配情况')



按照卫星厅和航站楼登机口画出登机口的使用数目和登机口的平均使用率

In [29]:

```
s_gates = [airport['gate'] for airport in airports if 'S' in airport['gate']]  
t_gates = [airport['gate'] for airport in airports if 'T' in airport['gate']]
```

In [30]:

```
s_gates_assign = {}; t_gates_assign = {}  
for gate in assign_dict.keys():  
    if gate in s_gates:  
        #print(len(assign_dict[gate]))  
        s_gates_assign[gate] = len(assign_dict[gate])  
    else:  
        #print('narrow'+str(len(assign_dict[gate])))  
        t_gates_assign[gate] = len(assign_dict[gate])
```

In [31]:

```
s_gates_assign = pd.DataFrame(s_gates_assign, index=[0]).T  
t_gates_assign = pd.DataFrame(t_gates_assign, index=[0]).T
```

In [32]:

```
plt.figure(figsize=(20, 10))
x = list(t_gates_assign.index)
plt.bar(x, t_gates_assign[0]*2, facecolor='b')
plt.xlabel('航站楼登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('航站楼登机口航班分配情况', fontsize=18)
plt.show()
```

Out[32]:

<Figure size 1440x720 with 0 Axes>

Out[32]:

<BarContainer object of 25 artists>

Out[32]:

Text(0.5,0,'航站楼登机口')

Out[32]:

Text(0,0.5,'每个登机口分配航班数量')

Out[32]:

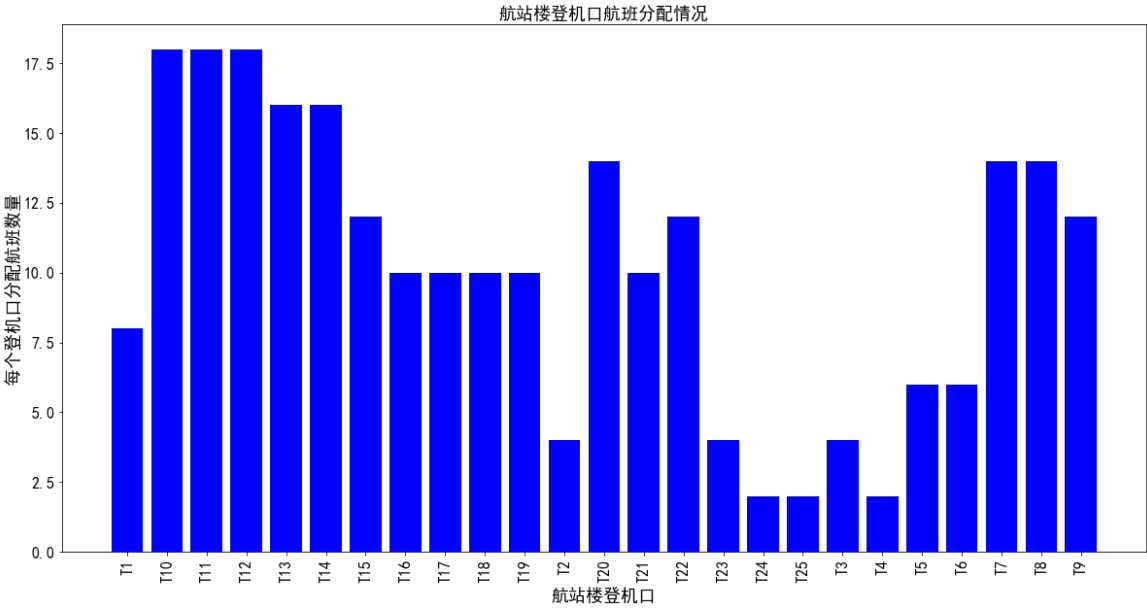
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24],
<a list of 25 Text xticklabel objects>)
```

Out[32]:

```
(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. ]),
<a list of 9 Text yticklabel objects>)
```

Out[32]:

Text(0.5,1,'航站楼登机口航班分配情况')



In [33]:

```
plt.figure(figsize=(20, 10))
x = list(s_gates_assign.index)
plt.bar(x, s_gates_assign[0]*2, facecolor='b')
plt.xlabel('卫星厅登机口', fontsize=18)
plt.ylabel('每个登机口分配航班数量', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('卫星厅登机口航班分配情况', fontsize=18)
plt.show()
```

Out[33]:

<Figure size 1440x720 with 0 Axes>

Out[33]:

<BarContainer object of 38 artists>

Out[33]:

Text(0.5,0,'卫星厅登机口')

Out[33]:

Text(0,0.5,'每个登机口分配航班数量')

Out[33]:

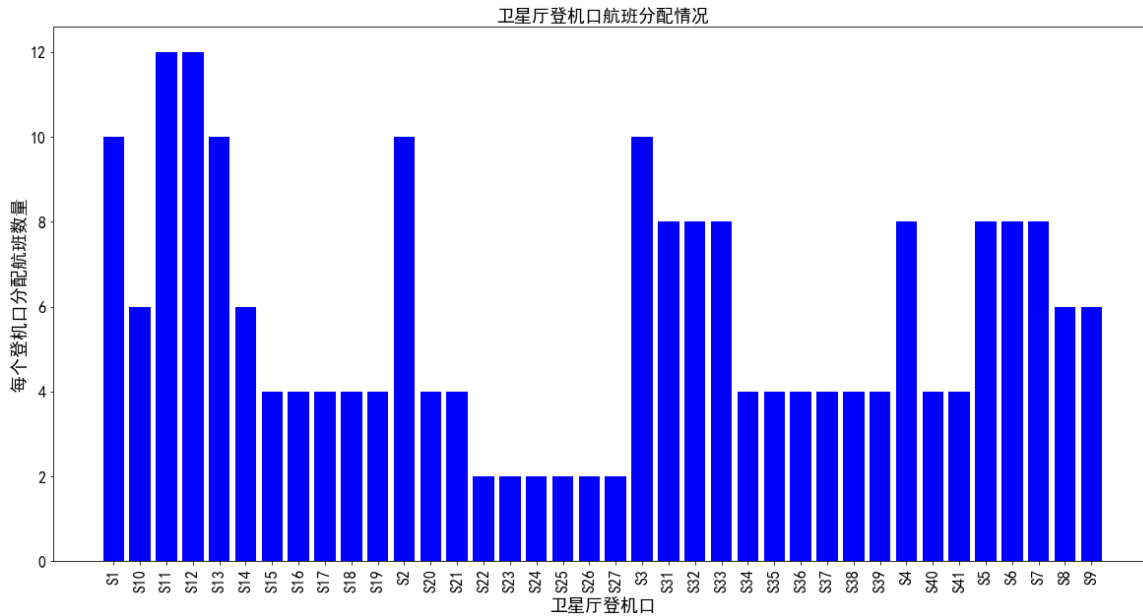
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37],
<a list of 38 Text xticklabel objects>)
```

Out[33]:

```
(array([ 0.,  2.,  4.,  6.,  8., 10., 12., 14.]),
<a list of 8 Text yticklabel objects>)
```

Out[33]:

Text(0.5,1,'卫星厅登机口航班分配情况')



In [34]:

```
s_airports = [airport for airport in airports if 'S' in airport['gate']]
t_airports = [airport for airport in airports if 'T' in airport['gate']]
```

In [35]:

```
s_busy_ratio = {}; t_busy_ratio = {}
assign_airport = list(assign_dict.keys())
for s_airport in s_airports:
    if s_airport['gate'] in assign_airport:
        all_time = len(s_airport['busy_time'])
        num_pucks = len(s_airport['puck_records'])
        busy_ratio = np.round((np.sum(s_airport['busy_time']) - 9 * num_pucks) /
all_time, 4)*100
        s_busy_ratio[s_airport['gate']] = busy_ratio

for t_airport in t_airports:
    if t_airport['gate'] in assign_airport:
        all_timet = len(t_airport['busy_time'])
        num_pucks = len(t_airport['puck_records'])
        busy_ratio = np.round((np.sum(t_airport['busy_time']) - 9*num_pucks) / al
l_timet, 4)*100
        t_busy_ratio[t_airport['gate']] = busy_ratio
```

In [36]:

```
s_busy_ratio = pd.DataFrame(s_busy_ratio, index=[0]).T  
t_busy_ratio = pd.DataFrame(t_busy_ratio, index=[0]).T
```

In [37]:

```
np.mean(s_busy_ratio)  
np.mean(t_busy_ratio)
```

Out[37]:

```
0    53.910789  
dtype: float64
```

Out[37]:

```
0    48.5412  
dtype: float64
```

In [38]:

```
plt.figure(figsize=(20, 10))
x = list(s_busy_ratio.index)
plt.bar(x, s_busy_ratio[0], facecolor='b')
plt.xlabel('卫星厅登机口', fontsize=18)
plt.ylabel('每个登机口使用率(%)', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('卫星厅登机口使用情况', fontsize=18)
plt.show()
```

Out[38]:

<Figure size 1440x720 with 0 Axes>

Out[38]:

<BarContainer object of 38 artists>

Out[38]:

Text(0.5,0,'卫星厅登机口')

Out[38]:

Text(0,0.5,'每个登机口使用率(%)')

Out[38]:

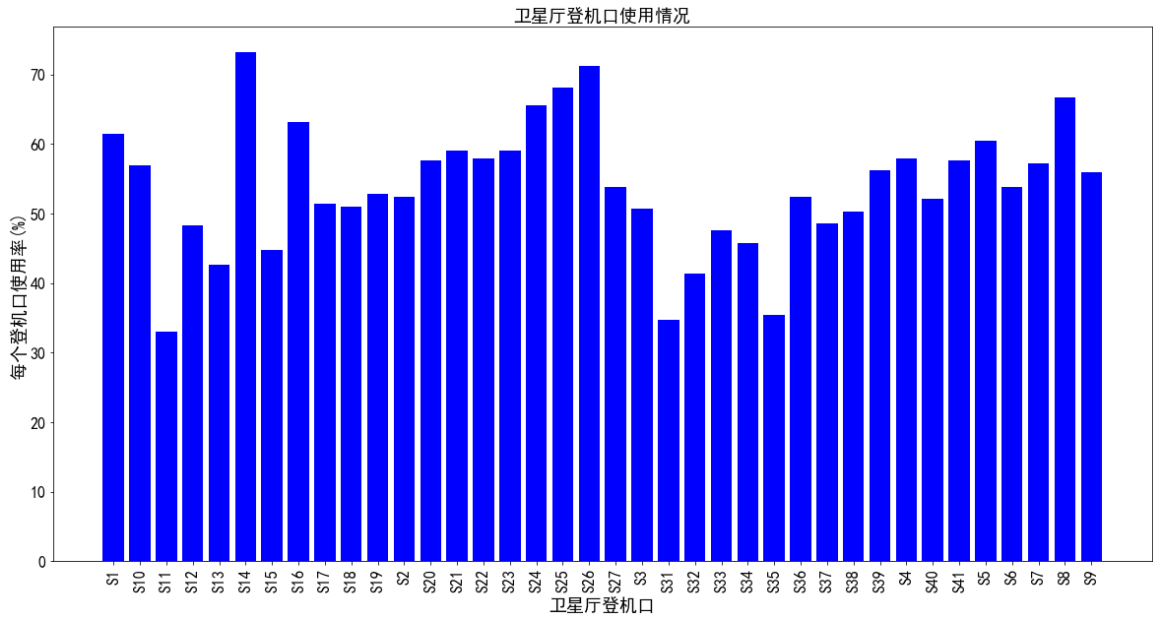
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37],
<a list of 38 Text xticklabel objects>)
```

Out[38]:

```
(array([ 0., 10., 20., 30., 40., 50., 60., 70., 80.]),
<a list of 9 Text yticklabel objects>)
```

Out[38]:

Text(0.5,1,'卫星厅登机口使用情况')



In [39]:

```
plt.figure(figsize=(20, 10))
x = list(t_busy_ratio.index)
plt.bar(x, t_busy_ratio[0], facecolor='b')
plt.xlabel('航站楼登机口', fontsize=18)
plt.ylabel('每个登机口使用率(%)', fontsize=18)
plt.xticks(rotation = 90, fontsize=16)
plt.yticks(fontsize=16)
plt.title('航站楼登机口使用情况', fontsize=18)
plt.show()
```


Out[39]:

<Figure size 1440x720 with 0 Axes>

Out[39]:

<BarContainer object of 25 artists>

Out[39]:

Text(0.5,0,'航站楼登机口')

Out[39]:

Text(0,0.5,'每个登机口使用率(%)')

Out[39]:

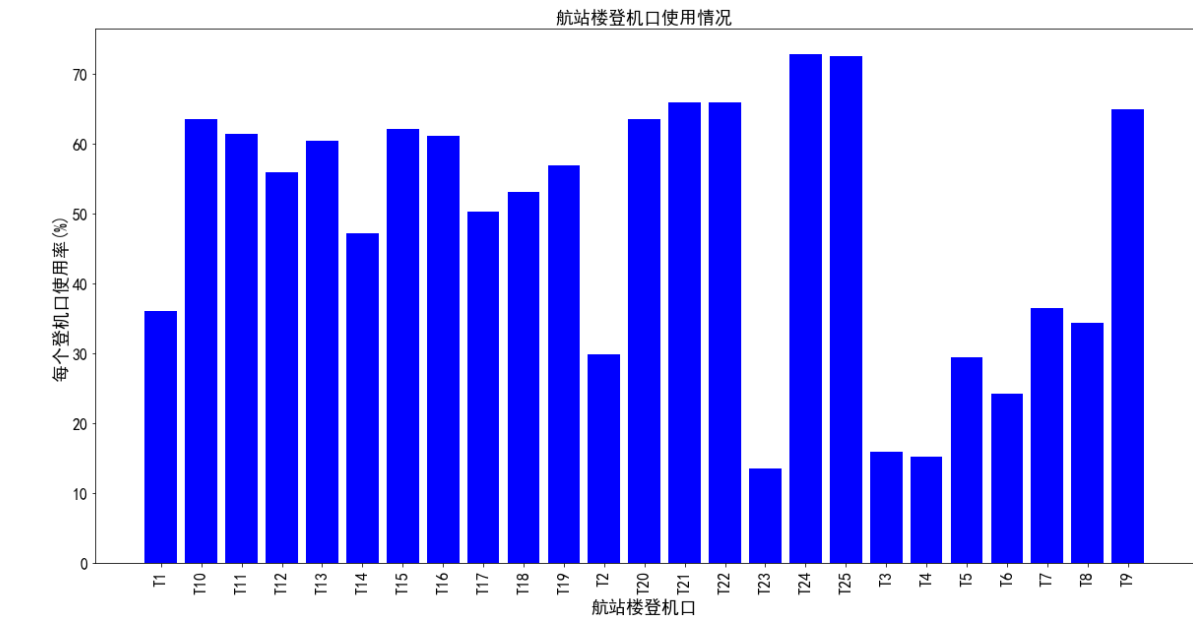
```
([0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
19,
20,
21,
22,
23,
24],
<a list of 25 Text xticklabel objects>)
```

Out[39]:

```
(array([ 0., 10., 20., 30., 40., 50., 60., 70., 80.]),
<a list of 9 Text yticklabel objects>)
```

Out[39]:

Text(0.5,1,'航站楼登机口使用情况')



计算中转失败乘客数量和比例

In [40]:

```
all_assign_pucks = [] # 所有已经分配的飞机转场记录号
for pucks in assign_pucks:
    all_assign_pucks.extend(pucks)
```

In [41]:

```
# assign_arrive_flights = []
# assign_depart_flights = []
# for puck in allpucks:
#     if puck['record'] in all_assign_pucks:
#         assign_arrive_flights.append(puck['a_flight'])
#         assign_depart_flights.append(puck['de_flight'])
```

In [42]:

```
tickets = pd.read_csv('./tickets_pass_totoal (1).csv')
tickets.head()
```

Out[42]:

	到达 航班	出发 航班	到达类型	出发类型	到达转场号	出发转场号	乘客数
0	NV677	NV6154	I	I	PK449	PK156	49
1	NV323	NV318	I	I	PK260	PK464	44
2	NV883	NV6773	I	D	PK466	PK488	30
3	NV343	NV6152	I	I	PK460	PK450	27
4	NV6155	NV656	I	I	PK285	PK304	26

In [43]:

```
def is_assign_pucks(puck):
    if puck in all_assign_pucks:
        return 1
    else:
        return 0
```

In [44]:

```
tickets['到达分配登机口'] = tickets['到达转场号'].map(is_assign_pucks)
tickets['出发分配登机口'] = tickets['出发转场号'].map(is_assign_pucks)
```

In [45]:

```
tickets['均分配到登机口'] = (tickets['到达分配登机口'] & tickets['出发分配登机口'])
```

In [46]:

```
compute_tickets = tickets[tickets['均分配到登机口']==1]
```

In [47]:

```
compute_tickets.reset_index(drop=True, inplace=True)
```

In [48]:

```
compute_tickets.head()
```

Out[48]:

	到达 航班	出发 航班	到达 类型	出发 类型	到达转 场号	出发转 场号	乘客 数	到达分配 登机口	出发分配 登机口	均分配到 登机口
0	NV323	NV318	I	I	PK260	PK464	44	1	1	1
1	NV883	NV6773	I	D	PK466	PK488	30	1	1	1
2	NV343	NV6152	I	I	PK460	PK450	27	1	1	1
3	NV6155	NV656	I	I	PK285	PK304	26	1	1	1
4	NV871	NV6152	I	I	PK455	PK450	26	1	1	1

In [55]:

```
tmpsum=tickets[(tickets.到达分配登机口==0)&(tickets.出发分配登机口==1)][ '乘客数' ].sum()
tmprate=tmpsum/2833
print(tmpsum)
print(tmprate)
```

592

0.2089657606777268