

Flight connecting problem based on zero-one integer programming

Zhiheng Jiang

Civil Aviation University of China, Tianjin, 300300, China

200440570@cauc.edu.cn

Abstract. With the rapid development of the civil aviation industry, the quantity of flights has significantly increased. However, how to take good advantage of airplanes becomes a problem, and this problem is the flight connection problem. Some flights can use one plane, while some flights cannot. Appropriate allocation of airplanes can maximize the profit and save the time. This paper chooses 12 pairs in Indira Gandhi National Airport. Then, it draws a bipartite graph and builds a zero-one integer programming model. Next, it introduces the constraints subject to the regulations in the aviation industry. Employing the branch and bound method is infeasible at the first time, which means some flights cannot be connected. After altering the constraints, an optimal solution is obtained, ensuring that the number of aircraft is minimal and the airport operates well at the same time. This flight connecting model can also be applied in other circumstances.

Keywords: Integer Programming, Flight Connection, Simplex Algorithm.

1. Introduction

Integer programming dates back to 1958, when Ralph Gomory issued the cutting plane method, the theoretical foundation of integer programming. Several years later, Land Doig, Dakin, and some other scholars came up with a renowned method, the branch and bound method. Today integer programming is very practical in many areas. It is widely used in graph theory to solve maximum-weight matching problems, minimum vertex cover problems, maximum independent set problems, and so on. It plays a key role in detecting optimal turning points in the financial area. In electrical power and energy systems, it can be applied to short-term power generation scheduling [1]. It can also be used in artificial intelligence areas to generate the optimal trajectory of a drone for wheelchair tracking [2]. In computer science, it can play a key role in transaction equivalence for privacy preservation [3]. It is also helpful in solving isolated community evacuation problems [4]. In energy systems, integer programming can be applied in the optimization of a district cooling network [5]. It can be divided into three categories: pure integer linear programming, which means all variables are integers; and mixed integer linear programming, which means some of the variables are integers. and zero-one integer linear programming, which means all variables are 0 or 1 [6].

The scale of the development in the civil aviation industry has been increasingly larger. A good flight scheme will minimize the number of aircraft to reduce the cost and increase the profit. In addition, flight time resources are extremely valuable in the operations of a hub airport [7]. And an appropriate flight schedule can dramatically save this kind of resource. Thus, there are a large number of research

concerning this. Chen employs a connection quality index calculation method to guide the airlines to coordinate their service schedules in Beijing Capital Airport and Beijing Daxing Airport and evaluate the flight connection, assuring that the flight connection in these airports is efficient [8]. Yaakoubi utilizes machine learning classification algorithms to generate a large-scale commercial solver (GENCOL), increasing annual revenue by dozens of millions of dollars in a large airline [9]. Riyadh Alhassan Riyadh Alhassan uses descriptive statistics, ARIMA (box and Jenkins) procedure, and multiple regressions to develop a forecasting model, increasing the total revenue of Saudi Arabian Airlines in comparison with its competitors [10]. This paper took flights to Indira Gandhi National Airport as an example, selecting some typical flights from October 25 to October 28. It utilized these data to draw a bipartite graph with the assistance of Python and Visio. Then it built a mathematical model similar to the maximum matching model. Next, it used the pulp library in Python to solve the flight connecting problem.

2. Method

2.1. 0-1 Integer programming

Integer programming is a branch of linear programming, so it is necessary to figure out linear programming before learning integer programming. Linear programming is to get an optimal solution of objective function through linear equations or linear inequations. But sometimes there are too many variables, it is difficult to solve them without a systematic method. This systematic method is called the simplex algorithm. The detailed steps of the simplex algorithm are shown as follows. The first step is to transfer the original model to a standardized model, the next step is to find an initial basic feasible solution. After that, in order to figure out which variable leaves and which enters, pivot steps are needed. The optimal solution will be obtained as long as all the coefficients of basic variables are negative. It is worth noting that not all of the models have a unique optimal solution. The solution can be unbounded, not unique, or infeasible. It can be concluded in Table 1:

Table 1. The step of the simplex algorithm.

c_j			c_1	...	c_m	...	c_j	...	c_n
C_B	x_B	b	x_1	...	x_m	...	x_j	...	x_n
c_1	x_1	b_1	1	...	0	...	a_{1j}	...	a_{1n}
c_2	x_2	b_2	0	...	0	...	a_{2j}	...	a_{2n}
...
c_m	x_m	b_m	0	...	1	...	a_{mj}	...	a_{mn}
$c_j - z_j$			0	...	0	...	$c_j - \sum_{i=1}^m c_i a_{ij}$...	$c_n - \sum_{i=1}^m c_i a_{in}$

The mathematic model is:

$$\begin{aligned}
 & \text{Max } Z = \sum c_j x_n \\
 & \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n \leq b_m \\ x_j \geq 0, 1 \leq j \leq n \end{cases} \quad (1)
 \end{aligned}$$

Where x_B , the set of elements $x_1, x_2, x_3, \dots, x_m$, is called the basis. The elements in the basis are basic variables and the remaining variables are non-basic variables. The unique optimal solution is obtained

only if two conditions are satisfied, that is, $c_j - z_j = 0$ for all of the basic variables and $c_j - z_j \leq 0$ for all of the non-basic variables. If the two conditions are not satisfied, one of the basic variables has to leave and one of the non-basic variables has to enter. Employing pivot steps with pivot rules is effective to ensure the two conditions are satisfied.

The simplex algorithm cannot ensure that the optimal solutions are integers. But in many cases, the solution will be infeasible if it is a decimal. Thus, it is necessary to add some conditions so that the solution will be integers. The branch and bound method is one of the most effective methods.

Let x_i be an arbitrary solution in the optimal solution set X_B . Assuming that $x_i = k_i$ are not integers, the original problem will be divided to two branches, one adds the constraint that $x_i \leq [k_i]$, where $[k_i]$ means the greatest integer less than k_i . The other part adds the constraint that $x_i \leq [k_i] + 1$. Then there will be two optimal solution sets. If neither of the sets is all integers, it will be divided again into two new branches and added aforementioned constraints for each previous branch. The increase of branches will not end unless all the solutions in X_B are integers.

2.2. Data analysis

Indira Gandhi National Airport is the heaviest airport in India with tremendous flights every year. Every flight needs flight crews and an aircraft, which brings huge costs. Thus, there will be a remarkable increase in profit if the airplanes and crews are minimized and the airport operates well at the same time. Because many flights in Delhi depart almost at the same time for every time period in one day, which means that there is no way to reuse the same airplane, this paper selects 12 typical Pairs in Delhi airport from October 25th, 2020, to October 28th, 2020 in Indira Gandhi National Airport, Delhi, which is shown in table 2.

Table 2. The departure times and arrival times of 12 pairs.

Delhi	Departure time		Arriving time	
Pair1	2020/10/25	5:05	2020/10/25	21:50
Pair2	2020/10/25	5:50	2020/10/25	23:55
Pair3	2020/10/26	6:10	2020/10/26	10:55
Pair4	2020/10/26	6:50	2020/10/26	11:55
Pair5	2020/10/26	7:00	2020/10/26	12:00
Pair6	2020/10/26	14:55	2020/10/26	13:30
Pair7	2020/10/26	15:35	2020/10/26	18:25
Pair8	2020/10/26	16:40	2020/10/26	22:40
Pair9	2020/10/26	17:10	2020/10/26	22:50
Pair10	2020/10/27	18:50	2020/10/28	6:00
Pair11	2020/10/27	19:55	2020/10/28	6:55
Pair12	2020/10/27	21:45	2020/10/28	7:15

2.3. Data processing

It is also indispensable to have a list consisting of time intervals between Pairs. However, it is time-consuming and ineffective to get that by manual calculation. So this table is imported to Python, and with the utilization of “numpy” library and “pandas” library, the time interval list is obtained as follows.

[3-1: 500, 4-1: 540, 5-1: 550, 3-2: 375, 4-2: 415, 5-2: 425, 6-3: 240, 7-3: 280, 8-3: 345, 9-3: 375, 6-4: 180, 7-4: 220, 8-4: 285, 9-4: 315, 6-5: 175, 7-5: 215, 8-5: 280, 9-5: 310, 7-6: 125, 8-6: 190, 9-6: 220, 10-7: 1465, 11-7: 1530, 12-7: 1640, 10-8: 1210, 11-8: 1275, 12-8: 1385, 10-9: 1200, 11-9: 1265, 12-9: 1375]. Where “3-1: 500” means the time interval between Pair 3 and Pair 1, the same goes for the rest.

It is necessary to note that the time interval between Pair 1 and Pair 6 is not calculated because the interval is so long that Pair 1 and Pair 6 cannot be connected. Others like Pair 1 and Pair 7, Pair 3 and Pair 11 are not connected due to the same reason

3. Flight connecting problem

3.1. Definition and Qualification

3.1.1. Definition. A plane can be used on many flights. For example, assuming that there are four flights: flight 1 means plane flying from destination A to B, flight 2 means plane flying from destination B to C, and flight 3 means flying from C to A. If flight 1, flight 2, and flight 3 satisfy that the time of arriving B in flight 1 is less than the time of departing from B in flight 2 and the time of arriving C in flight 2 is less than the time of departing from C, then the three flights can be integrated into one set. This set is called Pair. Theoretically, one plane is enough to finish all these flights.

Similarly, a Through is a combination of a series of aforementioned Pairs. One plane is enough to finish a Through.

3.1.2. Qualification. To ensure the mathematic models can be established, there must be some qualifications.

First, airplanes need time to finish tasks such as loading cargo, adding fuel, and maintenance. So it is impossible for an airplane to depart immediately after landing and stopping on the ground. Ordinarily, it takes 45 minutes to finish the obligatory tasks and prepare for the next flight. The two Pairs cannot be linked up until the difference between the arrival time of the preceding Pair and the departure time of the following Pair is more than 45 minutes. Sometimes the upper bound of the time interval is also restricted because the flight crews cannot wait too long, but in this circumstance, the intention is to minimize the total time. Thus, it is not indispensable to set the upper bound.

Second, the departing time and arriving time here are the times of leaving the stand and stopping at the stand respectively. And the delay is not be considered.

Then, as long as the time and places are appropriate, one airplane can be utilized on different flights, regardless of which airline it belongs to. For example, assuming that there are two flights, Delhi to Mumbai and Mumbai to Delhi, if the time of arriving in Mumbai is more than 45 minutes longer than departing from Mumbai to Delhi, the airplane in flight Delhi to Mumbai can also be used in flight Mumbai to Delhi, although these two flights might belong to different airlines.

3.2. Problem Description

Time cost and economic cost are two of the most important costs, so both of them need to be considered when setting objective functions. The coefficient in the objective function can be the differences between the arrival time of one Pair and the departure time of another Pair which is linked behind. By minimizing the objective function, the time cost can be reduced as possible as it can be. Minimizing the number of airplanes is equal to maximizing aircraft utilization. So it can be achieved through setting every possible constraint equal to one. If the model is infeasible after calculation, it means that some constraints cannot be satisfied. Thus, the constraints need to be altered, meaning that there is no way to link the same Pairs but add an airplane. Otherwise, it shows that all of the Pairs can be linked by choosing the correct scheme. In order to make the scheme feasible, the arrival time of preceding Pairs less than the departure time of the following Pairs must be ensured and the 45-time interval needs to be considered as well. In addition, the set consists of preceding Pairs, and the set made of following Pairs are one-to-one mapping. In other words, for every former Pair, it can only link one Pair behind. And for every latter Pair, it can only have one Pair precedes it. The detailed possible linking methods are demonstrated in Figure 1, where 1 means Pair 1 mentioned before in Table 2 and so as the others. The number "1" in both sides mean the same Pair, Pair 1. Others are the same. It is a bipartite graph.

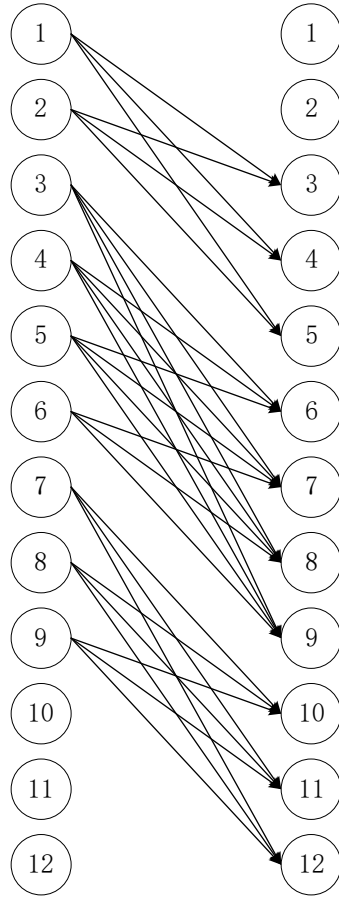


Figure 1. The bipartite graph of flight connection problem.

This bipartite graph has already eliminated lines of Pairs that cannot be linked due to excessive time intervals and infeasibility.

3.2.1. Mathematic model. The construction of mathematical model is not difficult when the problem description is clear. And according to integer programming, it can be set like this:

$$\begin{aligned} \text{Min} Z &= C_{ij} * X \\ \text{s.t. } Ax &\geq 1 \\ x &\in \{0,1\} \end{aligned} \quad (2)$$

Where C_{ij} means the time interval between different Pairs, constraint $Ax \geq 1$ means each Pair can only be matched once. If x equals to one, that means the two Pairs are matched, if not, means the two Pairs are disconnected.

It is similar to the maximum matching problem, but in maximum matching, the constraints are “no greater than one” instead of “no less than one”. Here the objective function needs to be minimized to ensure minimal time consumption, but the maximum matching needs to maximize the number of matches.

3.2.2. Solution. Unfortunately, it is infeasible to get an optimal solution. After checking the simplex tableau, it is found that both Pair 1 and Pair 2 do not choose Pair 5 as the following Pair. And it is easy to make sense, because no matter Pair 1 or Pair 2, the time interval that choosing Pair 5 as the following one is longest regardless of choosing Pair 3 or choosing Pair 4. So the constraints need to be modified

and an optimal solution can be obtained if the constraint " $x_{15} + x_{25} \geq 1$ " turned to " $x_{15} + x_{25} \leq 1$ ". The optimal basic solution is showed in Figure 2.

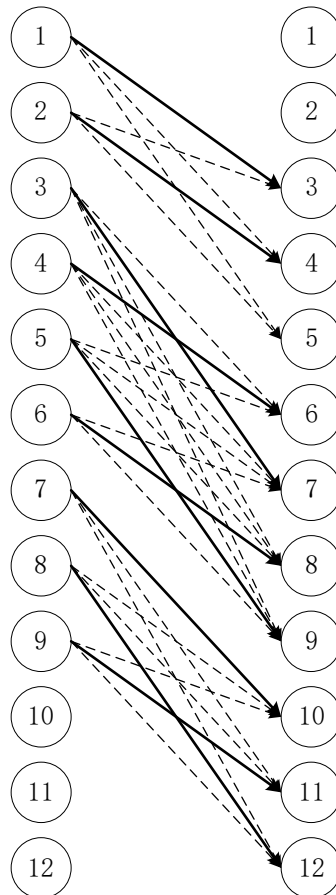


Figure 2. The final connecting scheme.

So the flight scheme for Indira Gandhi National Airport can be demonstrated in Table 3.

Table 3. The connection scheme.

Through 1	Pair 1	Pair 3	Pair 7	Pair 10	
Through 2	Pair 5				
Through 3	Pair 2	Pair 4	Pair 6	Pair 8	Pair 12

It is clear to see that the minimum number of aircraft is three.

4. Conclusion

This paper selects 12 typical Pairs and uses Python to get the time interval between different Pairs. Then, it draws a bipartite graph and builds the mathematic model. Next, with the help of pulp library in python and the branch and bound method, the problem is solved. However, it shows there is no feasible solution. Thus, the constraints have to be altered, which means two aircrafts are not enough to finish all the Pairs and it is necessary to get an extra airplane. Here is the final scheme: Pair 1, Pair 3, Pair 7 and Pair 10 are made of a Through, Pair 5 is a Through and Pair 2, Pair 4, Pair 6, Pair 8 and Pair 12 consist a Through.

Due to the lack of data about the crew members, this paper did not consider the arrangement of the crew members. It will be more practical if the allocation of crew members is taken into consideration because flight crews need time to take a break after finishing a flight task. Besides, this model did not

take the delay into consideration. In the future, it is better to introduce a delay prediction model, which means it can calculate the probability of a pair that delays and the amount of time it takes for the delay. In that case, the constraints will alter slightly according to the delay prediction model. Apart from importing a model, this paper can employ other methods like the Hungarian algorithm to solve it and compare the efficiency between distinctive methods.

References

- [1] dos Santos K V, Colonetti B, Finardi E C, Zavala V M 2023 Int. J. Elec. Power. 145 108689
- [2] Watanabe S, Mukai M 2022 Artif. Life Robot. 27 1 159-164
- [3] Krishnamoorthy S 2022 Comput. Oper. Res. 148 105997
- [4] Krutein K F, Goodchild A 2022 Trans. Res. Part E Logis. Trans. Rev.161 102710
- [5] Neri M, Guelpa E, Verda V 2022 Appl. Energ. 306 117994
- [6] Yunquan H 2018 Beijing Tsinghua University Press
- [7] Xiong L, Xiaoqing C 2018 IOP Conf. Series Earth Envir. Sci. 189 6
- [8] Chen Y, Zhu X, Wang L 2022 Inter. Conf. Energ. Storage Intell. Vehi. 926-934
- [9] Yaakoubi Y, Soumis F, Lacoste Julien S 2020 arXiv
- [10] Riyadh A 2017 J. King Abdul. Uni. Engin. Sci. 28 1