Planning an Optimal Trip to Europe

John Gilbertson, Shian Su, Ria Szeredi and Kenneth Young

Abstract

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Keywords: personalised tourist guide, linear programming, transportation planning, trip generation, decision making

Contents

1 Introduction

Introduction to the business you have selected. Which activity of the business have you selected to improve on? Why?

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2 Data

For the sake of brevity, we shall only provide data sets for the case of five cities here. This will be enough to demonstrate how we stored and used our data. The full data set can be found in the appendicies. All values are in Australian Dollars.

Our model will need to know the flight costs between all the cities. Initially we assumed that flights were uniform across all days of the trip and all times of each day. Our flight data was sourced from `momondo.com.au` [1]. We searched for only the cheapest flights between any two cities which imposed more assumptions on our problem.

Table 1: Airfares between Five Cities

	Moscow	Paris	London	Madrid	Rome
Moscow	0	146	126	202	146
Paris	227	0	60	143	93
London	213	82	0	249	160
Madrid	188	86	136	0	144
Rome	223	80	125	146	0

For simplicity we assumed that you can fly directly between all cities, even though in some cases the cheapest flight was not direct. This assumption is vaild as we are not attempting to find the optimal route around Europe, but instead minimise the overall cost of the trip or maximise the enjoyment gained.

Aside from the cost of airfares, the cost of living is another contributor to our model's objective value. All our daily living expenses data were sourced from `budgetyourtrip.com` [2]. To simplify the computation of our daily costs, we have assumed that there is no reduction in cost if a traveller books accommodation at a single hotel over multiple nights.

Table 2: Daily Costs of Five Cities

	Moscow	Paris	London	Madrid	Rome
Low	36	81	110	57	68
Mid	92	223	298	148	169
High	233	657	845	393	423

The last contribution to our model's objective value came from the flights from and to Melbourne at the begining and end of the trip. We assumed that the traveller bought a return ticket, and so this will force the first and last city that they visit to be the same. This is a reasonable assumption to make as the cost of purchasing a return ticket was generally less than that of purchasing two separate one-way tickets. This flight data was again sourced from `momondo.com.au` [1].

We assumed that all flights occur in the morning. We make this simplification as if the flight was at midday then the daily cost would have contributions from the city the traveller was leaving and the city they are going to. Further to this, we assumed that the return flight to Melbourne occurred on the morning after the last day of the trip. This was

Table 3: Return Flight Costs of Five Cities

Moscow	Paris	London	Madrid	Rome
2311	2155	1908	1870	2584

again to simplify the calculation of daily costs. If the traveller leaves a city in the early morning then we assume that there is no living expenses for that day.

Table 4: Base Utilities of Five Cities

Moscow	Paris	London	Madrid	Rome
71	93	100	77	85

The true utility of staying in a city is highly subjective and very difficult to measure. So for a basic model our utility is based on the overall popularity of a city as measured by a "bednights" statistic, which measures the number of nights tourists stayed in a given city. Because the difference in the raw data was quite large, the fifth root was taken such that all values were of similar magnitude and then rescaled to be a rounded percentage of the highest utility city. This reflects the behaviour of a naive traveller who makes their decisions on general popularity, later we explore specific preferences.

3 Solution Methodology

3.1 Heuristics

This problem has many similarities with the Travelling Salesman Problem (TSP), which is known not to be computationally efficient to solve. To avoid this issue, it can be useful to solve the problem using a greedy heuristic.

3.1.1 Cheap Heuristic

The cheap heuristic attempts to find the cheapest 15 day trip with 15 possible cities. For each city, the heuristic calculates the cost of travelling from Melbourne to that city and remaining in that city for maxDays. The algorithm then chooses the city that has the minimum cost. In the same way, cities are added recursively until there are no days left. The traveller then flies back to Melbourne from their final city.

Algorithm 1 Cheap Heuristic

```
Begin in Melbourne
for each city i do
    cost(i) = costFromMelb(i) + maxDays × costDaily(i)
end for
Go to city i with the minimum cost and stay for maxDays
daysLeft = days − maxDays
while daysLeft > 0 do
    step = min(maxDays, daysLeft)
    for each city i not yet visited do
        cost(i) = costTravel(currentCity, i) + step × costDaily(i)
    end for
    Go to city i with the minimum cost and stay for step days
    Decrement daysLeft by step
end while
Return to Melbourne from final city
```

Using maxDays = 4 and the medium daily costs defined in Section DATA, the output of this heuristic is shown in Table 5. The heuristic solution is to spend the first 4 days in Istanbul, then fly to Moscow for 4 days, followed by Prague for 4 days and finally Venice for 3 days. The cost of this trip is $4,060 including Melbourne flights, flights within Europe and daily costs. The utility of this trip is $1,102$.

3.1.2 Maximum Utility Heuristic

The maximum utility heuristic attempts to find the 15 day trip with the maximum utility regardless of cost. At each iteration, the algorithm chooses the city with the maximum utility from the set of cities containing the unvisited cities and the current city. After

Number of Days	City
4	Istanbul
4	Moscow
4	Prague
3	Venice

Table 5: Cheap Heuristic Output

spending a day in any city, the utility of staying another day in that city is multiplied by a decay factor.

Algorithm 2 Maximum Utility Heuristic

Begin in Melbourne
for $j = 1, \ldots,$ days **do**
  Find city i with the maximum utility in the set $\{\text{unvisited cities}\} \bigcup \{\text{current city}\}$
  Go to city i
  Reduce the utility of the current city by the decay factor
end for
Return to Melbourne from final city

Using a decay factor of 0.98 and the medium daily costs defined in Section DATA, the output of this heuristic is shown in Table 6. The utility of this trip is $1,330$, which is 21% greater than the utility of the trip found using the cheap heuristic. The cost of this trip is $\$5,450$, which is 34% ($\$1,390$) greater than the cheap heuristic.

Number of Days	City
4	London
4	Paris
2	Berlin
4	Rome
1	Barcelona

Table 6: Maximum Utility Heuristic Output

3.2 Mixed-Integer Linear Program

3.2.1 Base model

Variables
x_{id}: [binary] equal to 1 if the traveller is in city i on day d.
y_i: [binary] equal to 1 if the traveller ever visits city i.
m_{ijd}: [binary] equal to 1 if the traveller moves from city i to city j on day d.

Parameters
D: the total number of days.
$\mathcal{C}$: the set of cities available for visiting.
μ_{ij}: the cost to move from city i to city j.
δ_i: the daily cost of city i.
α: the minimum number of days allowed in a city.
ω: the maximum number of days allowed in a city.

$$\text{Min} \sum_{d=1}^{D} \sum_{i \in \mathcal{C}} \delta_i \cdot x_{id} + \sum_{d=1}^{D-1} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \mu_{ij} \cdot m_{ijd}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{d=1}^{D} x_{id} \geq \alpha \cdot y_i && \forall i \in \mathcal{C} && (1)\\
& \sum_{d=1}^{D} x_{id} \leq \omega \cdot y_i && \forall i \in \mathcal{C} && (2)\\
& x_{id} + x_{j(d+1)} \leq 1 + m_{ijd} && \forall i, j \in \mathcal{C}, d = 1..D-1 && (3)\\
& \sum_{i \in \mathcal{C}} x_{id} = 1 && d = 1..D && (4)\\
& \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} m_{ijd} = 1 && d = 1..D-1 && (5)\\
& \sum_{d=1}^{D} \sum_{j \in \mathcal{C}} m_{ijd} \leq 1 + \sum_{d=1}^{D} m_{iid} && \forall i \in \mathcal{C} && (6)\\
& x_{j1} + \sum_{d=1}^{D} \sum_{i \in \mathcal{C}} m_{ijd} \leq 1 + \sum_{d=1}^{D} m_{jjd} && \forall j \in \mathcal{C} && (7)\\
& x_{id}, y_i, m_{ijd} \in \{0, 1\} && \forall i, j \in \mathcal{C}, d = 1..D && (8)
\end{aligned}$$

The base model is a simple minimisation of cost during a trip through europe. It's clear that the optimal solution would be to simply fly to the cheapest city and stay there for the duration of the planned trip. So we set some limits on how long we can stay in a particular city.

3.2.2 Decaying Enjoyment

Assuming that the enjoyment of staying in a city decays with the number of days stayed. We add the following to the model.

Variables

s_{ij}: [binary] equal to 1 if traveller stays in city i for j days.

Parameters

u_{ij}: The total utility of staying in city i for j days.

Then we no longer need (2) since we hope that the traveller naturally leaves the city once the marginal enjoyment is low enough but we retain (1) since it's still reasonable to set a length of minimum stay.

Then the new model is

$$\text{Max} \sum_{i \in \mathcal{C}} \sum_{j=1}^{D} s_{ij} * u_{ij}$$

$$\text{s.t.} \quad \sum_{d=1}^{D} x_{id} \geq \alpha \cdot y_i \qquad \forall i \in \mathcal{C} \tag{9}$$

$$\sum_{j=1}^{D} s_{ij} = y_i \qquad \forall i \in \mathcal{C} \tag{10}$$

$$s_{ij} \cdot j \leq \sum_{d=1}^{D} x_{id} \qquad \forall i \in \mathcal{C}, j = 1...D \tag{11}$$

$$x_{id} + x_{j(d+1)} \leq 1 + m_{ijd} \qquad \forall i, j \in \mathcal{C}, d = 1..D-1 \tag{12}$$

$$\sum_{i \in \mathcal{C}} x_{id} = 1 \qquad d = 1..D \tag{13}$$

$$\sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} m_{ijd} = 1 \qquad d = 1..D-1 \tag{14}$$

$$\sum_{d=1}^{D} \sum_{j \in \mathcal{C}} m_{ijd} \leq 1 + \sum_{d=1}^{D} m_{iid} \qquad \forall i \in \mathcal{C} \tag{15}$$

$$x_{j1} + \sum_{d=1}^{D} \sum_{i \in \mathcal{C}} m_{ijd} \leq 1 + \sum_{d=1}^{D} m_{jjd} \qquad \forall j \in \mathcal{C} \tag{16}$$

$$x_{id}, y_i, m_{ijd} \in \{0, 1\} \qquad \forall i, j \in \mathcal{C}, d = 1..D \tag{17}$$

With this we find the optimal enjoyment possible for the traveller irrespective of cost, then we optimise again to minimise cost at the optimal enjoyment.

$$\text{Min} \sum_{d=1}^{D} \sum_{i \in \mathcal{C}} \delta_i \cdot x_{id} + \sum_{d=1}^{D-1} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \mu_{ij} \cdot m_{ijd}$$

s.t.

(1)-(9) from previous model is statisfied and

$$\sum_{i \in \mathcal{C}} \sum_{j=1}^{D} s_{ij} * u_{ij} = (\sum_{i \in \mathcal{C}} \sum_{j=1}^{D} s_{ij} * u_{ij})_{optimal}$$

We look at the new constraints introduced.

Constraint	Explanation
$\sum_{j=1}^{D} s_{ij} = y_i$ c	This ensures that only one stay duration is valid for each city, also that the stay durations are zero if a city is not visited. d
$s_{ij} \cdot j \leq \sum_{d=1}^{D} x_{id}$	When trying to maximise utility the program will try to set j to the highest value it can for each i because u_{ij} is set up as the cumulative utility of staying in city i for j days. As a result this sets s_{ij} 1 if we stay in city i for j days.

Now in practice, solving the model in Fico Xpress takes a substantial amount of time as the number of cities and days of the trip grows. So we impose some additional constraints, first we note that with unconstrained utility and costs, both the utility and costs are non-decreasing with the number of days, then we can solve a shorter trip and use its results as lower bounds for cost and utility in the longer trip. This improves the running time significantly for maximising utility, furthermore we note that for an optimal utility, the number of days in each city will not change in the optimal cost optimisation. Therefore we can use this as a constraint for improving the performance of the cost optimisation. Then we can add the constraint

$$\sum_{j=1}^{D} s_{ij} = (\sum_{j=1}^{D} s_{ij})_{optimal} \qquad \forall i \in C.$$

If we have multiple solutions that optimise utility then we can simply optimise cost for each of them and take the minimum.

4 Extensions and Results Discussion

Currently our base model takes limited input from the propective traveller; only the number of days of the trip and possibly the minimum and maximum number of days spent in any one city. To make the model more applicable in real world scenarios we propose some extensions which can handle more varied inputs.

As a brief example, let us consider a traveller who does not want to visit Istanbul. Adding the following set of constraints to the base model will impose the traveller's needs (Istanbul's index is 13),

$$x_{13,d} = 0 \quad , \quad \forall d \in D$$

The base model always returned Istanbul as one of the cities visited, as it had the overall cheapest flights. So the traveller's dislike of Istanbul has increased the minimal cost of their trip from \$3672 to \$3860.

4.1 Varying Airfares

So far, we have assumed that the cost of flights was uniform across all days. This extension's aim is to ensure our model returns more realistic results by considering how the cost of airfares vary across the week. We use `momondo.com.au` again to source the airfare data [1]. To reduce computation times we arbitrarily assume that the trip begins on a Sunday.

To model the variation in airfares, the travel cost parameter μ_{ij} needs an additional dimension to store the day of the week. So we now redefine μ as follows,

μ_{ijd}: The cost to fly from city i to city j on weekday d, $\forall i, j \in \mathcal{C}$, $d = 1...7$.

If the size of $\mathcal{C}$ is n, then we now have a 3-dimensional array of size $n \times n \times 7$. To define an array in the Mosel language, one must specify each element and its corresponding index tuple. There are many ways of shortcutting this process, however to define the elements of a 3-dimensional array the shorthand becomes rather inelegant. In the case of 15 cities, the parameter definition section became longer than the model definition itself.

Seeking an alternate and more compact implementation, we decided to take the current flight data as the base price to travel between any two cities. Then construct a new parameter which stored how the flight cost was augmented for each day of the week. So now, for each of the cities, we have a size $n \times 7$ array which stores these augmentation values as percentages,

Parameters

ai_{jd}: The percentage increase in airfares from city i to city j on day d, $\forall i, j \in \mathcal{C}$, $d = 1...7$.

To demonstrate this we provide Moscow's augmentation array, $a1_{jd}$. An entry of zero indicates that there is no increase in price for the flight on that day. The full table can be found in the appendicies.

Table 7: Airfare Augmentation Data for Moscow

City	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Moscow	0	0	0	0	0	0	0
Paris	4	1	0	3	8	13	10
London	11	3	0	2	6	8	6
Madrid	10	3	0	6	13	17	14
Rome	4	1	0	3	11	12	12

Now adding this to the objective function gives

$$\text{Min} \sum_{d=1}^{D}\sum_{i\in\mathcal{C}} \delta_i \cdot x_{id} + \sum_{d=1}^{D-1}\sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{C}} m_{ijd} \cdot (\mu_{ij(d \bmod 7)} + 0.01 \cdot \mu_{ij(d \bmod 7)} \cdot ai_{jd})$$

As an illustration, now consider a traveller planning a 12 day trip, who needs visit Florence and then travel to Vienna for the last 4 days of their trip. Adding the following constraints to the model,

$$x_{10,8} = 1$$
$$x_{14,d} = 1 \qquad \forall d = 9...12,$$

we receive the results from the base model and varying airfares extension as listed below.

Table 8: Base Model Output

Number of Days	City
x	X
x	X
x	X
4	Vienna

Table 9: Varying Airfares Output

Number of Days	City
x	X
x	X
x	X
4	Vienna

The base model returns a total cost of $XXX. Whereas with the variable airfare data added to the model, the minimum cost increases by $ZZZ (%XX.XX) to $YYY. This cost increase comes from the %35 increase in flights from Florence to Vienna on a Sunday and the %XX increase in flights from Vienna to Melbourne on a Thursday. So removing the assumption that flight costs are uniform across all days has given us a better estimate of the minimum cost.

4.2 Activity Preference

4.3 Avoiding or Joining Another Traveller

4.4 Multiple Person Trip

5 Conclusion and Recommendations

Write your conclusions here. Througout the whole document, do not forget to cite your sources [3].

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

6 Acknowledgements

Acknowledge anyone (person, organisation etc.) who has contributed to your project.

alysson for being such a sexy beast

7 Appendix

Daily expenses table goes here

Table 10: Airfares between Twenty Five Cities (Part I)

From\ To	Moscow	Paris	London	Madrid	Rome	Crete	Barcelona	Berlin	Budapest	Florence	Amsterdam
Moscow	0	146	126	202	146	168	143	156	207	291	154
Paris	227	0	60	143	93	123	139	110	162	85	52
London	213	82	0	249	160	284	135	107	163	224	124
Madrid	188	86	136	0	144	188	69	70	96	138	103
Rome	223	80	125	146	0	177	39	96	69	84	94
Crete	233	81	223	188	58	0	131	123	58	241	200
Barcelona	200	121	79	76	32	165	0	143	108	103	90
Berlin	130	53	81	97	83	153	110	0	83	214	92
Budapest	249	133	125	96	41	123	96	76	0	164	157
Florence	309	246	162	172	86	301	90	186	204	0	227
Amsterdam	155	158	73	141	136	211	117	106	125	229	0
Prague	112	122	100	112	48	193	116	108	138	144	97
Istanbul	123	101	127	154	67	173	122	86	41	235	66
Vienna	145	162	122	202	90	75	145	117	175	171	90
Venice	167	65	17	117	84	191	97	119	110	152	75
Goreme	172	206	205	237	137	253	181	142	86	522	136
Lisbon	282	80	77	50	135	220	107	123	117	173	55
Nice	278	127	119	143	48	220	50	90	138	173	121
Reykjavik	566	190	284	337	242	484	217	260	246	454	246
Edinburgh	233	194	71	135	204	371	179	115	151	390	168
Dublin	196	87	58	102	146	265	107	86	143	253	138
Krakow	191	135	94	126	142	286	107	176	138	181	143
Copenhagen	258	174	71	158	160	315	95	56	157	200	152
Athens	118	149	156	135	72	39	156	109	91	230	109
Munich	188	120	136	163	69	112	135	110	126	192	170

Table 11: Airfares between Twenty Five Cities (Part II)

From\ To	Prague	Istanbul	Vienna	Venice	Goreme	Lisbon	Nice	Reykjavik	Edinburgh	Dublin	Krakow
Moscow	146	115	131	129	167	283	200	406	266	263	236
Paris	165	150	140	68	375	178	114	240	124	106	152
London	171	197	179	152	267	205	160	183	116	116	177
Madrid	124	180	143	119	168	57	152	336	127	125	107
Rome	94	74	105	71	157	155	62	180	155	174	147
Crete	114	120	164	169	265	154	144	394	178	211	214
Barcelona	110	141	103	74	203	69	50	241	151	117	100
Berlin	123	99	176	104	197	118	108	245	100	102	128
Budapest	113	50	149	97	100	104	130	265	117	104	107
Florence	155	279	229	150	470	197	233	421	223	228	107
Amsterdam	85	121	155	95	200	137	73	252	88	101	134
Prague	0	85	152	65	186	174	135	256	134	114	141
Istanbul	108	0	59	65	33	140	126	337	143	162	149
Vienna	181	121	0	194	152	254	166	284	199	211	147
Venice	96	177	152	0	274	153	200	316	135	146	127
Goreme	206	27	55	230	0	257	130	384	189	198	440
Lisbon	178	260	161	123	441	0	66	328	142	108	174
Nice	96	180	157	128	224	116	0	313	175	97	189
Reykjavik	285	334	370	323	332	325	334	0	175	97	189
Edinburgh	140	294	370	127	613	174	168	192	0	39	139
Dublin	110	217	156	146	396	114	138	254	34	0	137
Krakow	180	247	197	205	362	171	139	341	133	119	0
Copenhagen	96	234	229	192	363	191	154	270	138	149	149
Athens	62	98	105	130	141	173	137	263	179	138	149
Munich	194	103	194	165	176	159	117	263	142	143	154

Table 12: Airfares between Twenty Five Cities (Part III)

From\ To	Copenhagen	Athens	Munich
Moscow	162	141	152
Paris	75	226	68
London	130	145	133
Madrid	108	169	184
Rome	143	160	28
Crete	176	46	144
Barcelona	104	146	93
Berlin	43	174	110
Budapest	123	133	122
Florence	164	268	158
Amsterdam	60	128	183
Prague	86	143	153
Istanbul	51	70	83
Vienna	177	154	149
Venice	184	181	101
Goreme	111	120	147
Lisbon	192	171	124
Nice	103	175	87
Reykjavik	218	175	266
Edinburgh	159	273	220
Dublin	60	186	155
Krakow	110	176	220
Copenhagen	0	128	218
Athens	131	0	145
Mucich	131	166	0

For brevities sake, we shall only include the tables detailing the airfare augmentation data for Moscow and Florence.

Table 13: Airfare Augmentation Data for Moscow

City	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Moscow	0	0	0	0	0	0	0
Paris	4	1	0	3	8	13	10
London	11	3	0	2	6	8	6
Madrid	10	3	0	6	13	17	14
Rome	4	1	0	3	11	12	12
Crete	2	0	0	2	7	13	5
Barcelona	12	3	0	7	13	16	14
Berlin	1	0	2	7	12	7	5
Budapest	0	2	2	7	10	7	5
Florence	5	0	5	15	19	15	7
Amsterdam	0	0	4	11	15	8	3
Prague	0	0	3	7	11	7	3
Istanbul	2	2	9	11	12	7	0
Vienna	0	0	7	12	16	14	7
Venice	2	0	5	14	22	14	6
Goreme	4	1	0	2	5	6	15
Lisbon	4	0	3	8	13	10	5
Nice	3	0	4	12	16	17	6
Reykjavik	1	0	6	4	6	7	1
Edinburgh	1	0	2	8	16	12	10
Dublin	6	0	3	10	20	17	15
Krakow	0	0	6	6	16	6	2
Copenhagen	0	0	3	6	10	10	10
Athens	2	0	3	6	13	15	3
Mucich	1	0	5	11	15	18	7

Table 14: Airfare Augmentation Data for Florence

City	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Moscow	20	7	0	5	18	28	39
Paris	20	6	0	5	18	25	39
London	17	0	0	10	12	13	40
Madrid	17	10	3	0	4	5	9
Rome	0	15	2	11	8	2	8
Crete	5	0	2	3	4	15	6
Barcelona	6	0	2	7	10	9	11
Berlin	18	7	0	13	10	27	28
Budapest	1	2	2	4	2	0	5
Florence	0	0	0	0	0	0	0
Amsterdam	4	3	2	5	0	12	18
Prague	5	0	4	7	3	4	5
Istanbul	4	0	1	3	4	7	8
Vienna	8	2	0	5	12	15	35
Venice	5	0	5	15	23	14	14
Goreme	5	2	0	1	4	6	10
Lisbon	12	0	1	8	12	11	13
Nice	5	3	2	0	1	1	9
Reykjavik	24	12	0	1	5	6	26
Edinburgh	10	0	3	8	15	16	25
Dublin	5	1	0	2	2	5	11
Krakow	26	9	3	0	0	5	11
Copenhagen	10	0	1	4	7	15	22
Athens	8	0	0	3	7	16	13
Mucich	6	5	5	0	4	6	20

References

[1] momondo.com.au, *Global Travel Search Engine*, accessed 6 May 2015. Available online: `http://www.momondo.com.au/`

[2] Budget Your Trip, *Trvavel Cost Search*, accessed 13 May 2015. Available online: `http://www.budgetyourtrip.com/`

[3] Foo, Bar, Smith (2002) *Example Title*. Example Publisher. ISBN XXX-XXX