RC24780 (C0904-002) April 9, 2009
Other

IBM Research Report

On the Tour Planning Problem

Chenbo Zhu, J. Q. Hu, Yifan Xu
Department of Management Science
School of Management
Fudan University
Shanghai 200433
P.R China

Fengchun Wang, Shun Jiang, Rongzeng Cao, Wei Ding
IBM Research Division
China Research Laboratory
Building 19, Zhouguancun Software Park
8 Dongbeiwang West Road, Haidian District
Beijing, 100094
P.R.China

On the Tour Planning Problem*

Chenbo Zhu[1], J.Q. Hu†[1], Yifan Xu[1]

Fengchun Wang[2], Shun Jiang[2], Rongzeng Cao[2], Wei Ding[2]

[1]Department of Management Science
School of Management, Fudan University
Shanghai 200433, China

[2]IBM China Research Laboratory
Beijing 100193, China

Abstract

Increasingly tourists are planning trips by themselves using the abundant information available on the Web, however they still expect and want trip plan advisory services. In this paper, we study the tour planning problem. Our goal in this problem is to design a tour trip of the most desirable sites, subject to various budget and time constraints. We first establish a framework for this problem, and then formulate it as a mixed integer linear programming problem. However, except when the problem size is relatively small, say, with less than 20-30 sites, it is computationally infeasible to solve the mixed-integer linear programming problem. Therefore, we propose a heuristic method based on local search ideas. The method is efficient and provides good approximation solutions. Numerical results are provided to validate the method.

1. Introduction

As a typical service industry, Travel & Tourism is one of the world's highest priority industries and employers ([1]). It encompasses transportation, accommodation, catering, recreation and services for visitors both at home and from overseas. Information Communication and Technologies have radically changed the efficiency and effectiveness of tourism organizations, the way that businesses catering to tourists operate, as well as how consumers interact with organizations ([2]). One of the major paradigm-shifts is due to the emergence of the Internet. A survey of US Web users conducted in 2007 by Burstmedia ([3]) shows that: four out of five (79.0%) respondents would use the Internet to plan their upcoming personal travel.

Generally speaking, most travel websites provide information, destination or travel package recommendation, online flight/hotel/car reservation, and community forums for sharing travel tips. With the help of such travel websites, a new type of user is emerging: they become their own travel agents and build their travel packages themselves ([4]). Trip planning is a complex constructive activity affected by various factors, which can be classified into two categories: 1) personal features, including both socioeconomic factors (such as age, education and income) and psychological and cognitive factors (such as experience, personality, and involvement), and 2) travel features (such as travel purpose, travel-party size, length of travel, distance, and transportation mode). In the study of the complex constructive activity, both researchers and practitioners have explored several possible approaches. One popular approach is to use recommendation systems based on Artificial Intelligence ([5]). In a Travel Recommender System (TRS), travelers' needs and constraints, through recommendation algorithms, are mapped into appropriate product selections based on the knowledge collected by the intelligent recommender. Knowledge can be extracted by using the following four approaches ([6]):

1. Non-personalized: recommending products to customers based on what other customers have said about the products on average;
2. Attribute based: recommending products to customers based on syntactic properties of the products;
3. Item-to-item correlations: recommending products to customers based on a small set of products the

* The work in this paper is supported in part by a grant from IBM China Research Laboratory and by the National Natural Science Foundation of China under Grant NSFC No. 70832002.

† Corresponding author: Tel.: 86-21-25011177; Fax: 86-21-65642412; E-mail: hujq@fudan.edu.cn

customers have expressed interest in;

4. People-to-people correlations: recommending products to a customer based on the correlation between that customer and other customers who have purchased products from the E-commerce site.

However, Ricci [7] argues that none of these approaches can support the user in building a “user defined” trip package that is consists of one or more tourist attractions, accommodations, caterings and other services. Current TRS can only support the first stage of trip package planning-destination selection, because: 1) a content based approach does not scale unless we pursue a costly knowledge-engineering activity, and 2) a collaborative-filtering approach encounters the complexity of the travel products: people can not simplify a trip to the point where two travelers’ trips are the same ([7]). Ricci and Wether [8] believe that a case based reasoning (CBR) approach might be more useful. Its basic idea is to load one previous case which best match the traveler’s personal data and preferences (room prices, type of accommodation, etc.) as a reuse base and then tailor it.

No matter what kind of recommendation systems are employed as advisors for tour package planning, a reference base would be extremely useful and can be used to simplify the selection process dramatically: the reference base could be either previous purchase/selection records by the user/other users or a prepackaged tour provided by a travel agency or other experienced travelers. Although some commercial websites claim that they can support tour trip planning, it’s more like an electronic notebook and organizer, which allows the user to record their selections of single products (e.g., a flight, a hotel, a car rental) in their trip planning. The user still has to plan their entire trip almost completely manually with little help from these websites.

In this paper, we aim to establish a framework for the tour planning problem, and then develop a mathematical model as well as efficient algorithms. In solving such a problem, our objective is to design a tour trip with the most desirable sites subject to various budget and time constraints. We will first formulate the problem as a mixed integer linear programming (MILP) problem. However, as in many similar applications, the tour trip planning MILP problem is NP-complete and it is computationally difficult to solve when the problem size is large.

The rest of the paper is organized as follows. In sections 2, we first introduce the tour planning problem and formulate the problem as a mixed-integer linear program problem. In section 3, we give a computational comparison between our results and exact results in an instance involving 33 sites to validate our algorithm, then a real instance with more sites is solved, and section 4 concludes.

2. Problem Formulation

Let $G = (V, E)$ be a connected graph with vertices $V = M \cup N$, where $M = \{1, \ldots, m\}$ represent m hotel sites and $N = \{m+1, \ldots, m+n\}$ represents n tourist sites, and edges $E = \{(i, j) : i, j \in V\}$. For ease of exposition, we assume that only one hotel is selected for the entire tour. However, our formulation and method can be extended to more general cases in which more than one hotel may be selected. We define:

1. C: the total budget available for the tour,
2. d: the number of days available for the tour,
3. T_k : the tour time available during the k th day ($k = 1, 2, \ldots, d$),
4. τ_i : the tour time that the tourist spends at site i, $i \in V$ ($\tau_i = 0$ for $i \in M$)
5. $[a_i, b_i]$: the time period during which the tourist is allowed to visit site i, $i \in V$, and we assume that it is the same for all days,
6. c_i : the cost associated with visiting (or staying at) site i, $i \in V$,
7. U_i : the utility of visiting (or staying at) site i, $i \in V$,
8. t_{ij} : the travel time between sites i and j, $i, j \in V$.

We note that one way to obtain the utility function for each site is to introduce a set of attributes, A (e.g., an attribute could be natural beauty, historical significance, or cultural heritage). Each attribute is assigned a weight w_j by the tourist, which indicates his preference to the attribute (the higher the weight, the more preferred the attribute by the tourist). Let $S = \{s_{ij} : i \in V, j \in A\}$, the attribute value set, where s_{ij} is the value of attribute j for site i. For example, if site i is a historical museum, then its value of the historical attribute is very high, the while its values for other attributes may be very small (or zero). Given S and W, we can obtain tourist's utility U_i for site i as: $U_i = \sum_{j \in A} w_j s_{ij}$.

The tour planning problem (TPP) is to construct a d-day tour that maximizes the total utility of all sites visited while satisfying the following constraints:

1. The number of days for the tour is d;
2. For each day, the tour starts and ends at the hotel;
3. Each tourist site is visited no more than once;
4. The arrival and the departure time at each tourist site is restricted by its permited visiting time $[a_i, b_i]$;
5. The cumulative tour time of the k th day does not exceed the available tour time T_k ;
6. The total cost of the d day tour does not exceed the budget C .

We now introduce the following decision variables:

$$x_{ijk} = \begin{cases} 1, & \text{if the tourist travels from site } i \text{ to site } j \text{ on the } k\text{th day,} \\ 0, & \text{otherwise.} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if site } i \text{ is visited by the tourist,} \\ 0, & \text{otherwise.} \end{cases}$$

$$z_i = \begin{cases} 1, & \text{if hotel } i \text{ is selected by the tourist,} \\ 0, & \text{otherwise.} \end{cases}$$

$t_i =$ the time epoch at which the tourist starts his visit at site i.

$t_{iak} =$ the time epoch at which the tourist departs from hotel i on the kth day.

$t_{ibk} =$ the time epoch at which the tourist arrives at hotel i on the kth day.

TPP can then be formulated as the following MILP problem:

$$
\begin{aligned}
\max \quad & U = \sum_{i\in N} U_i y_i + \sum_{i\in M} U_i z_i & & (1)\\
s.t. \quad & \sum_{i\in V} x_{irk} = \sum_{j\in V} x_{rjk} & r \in N, k \in D, \quad & (2)\\
& \sum_{i\in N} x_{irk} = \sum_{j\in N} x_{rjk} & r \in M, k \in D, \quad & (3)\\
& y_i = \sum_{k\in D}\sum_{j\in V} x_{ijk} & i \in N, \quad & (4)\\
& z_i = \frac{1}{d}\sum_{k\in D}\sum_{j\in N} x_{ijk} & i \in M, \quad & (5)\\
& \sum_{i\in M} z_i = 1 & & (6)\\
& t_i + \tau_i + t_{ij} - (1 - x_{ijk})T_k \le t_j & i, j \in N, k \in D, \quad & (7)\\
& t_{iak} + t_{ij} - (1 - x_{ijk})T_k \le t_j & i \in M, j \in N, k \in D, \quad & (8)\\
& t_i + \tau_i + t_{ij} - (1 - x_{ijk})T_k \le t_{jbk} & i \in N, j \in M, k \in D, \quad & (9)\\
& a_i \le t_i \le b_i & i \in N, \quad & (10)\\
& a_k \le t_{iak} & i \in M, k \in D, \quad & (11)\\
& t_{ibk} \le b_k & i \in M, k \in D, \quad & (12)\\
& \sum_{i\in N} c_i y_i + d \sum_{j\in M} c_j z_j \le C & & (13)\\
& x_{ijk} \in \{0,1\} & i, j \in V, k \in D, \quad & (14)\\
& y_i \in \{0,1\} & i \in N, \quad & (15)\\
& z_i \in \{0,1\} & i \in M. \quad & (16)
\end{aligned}
$$

The objective function (1) maximizes the total utility of all sites visited and constraint (2)-(16) are:

- Constraint (2) is the balance equation, which ensures that if the tourist visits a site, he would leave the site.
- Constraint (3) ensures that the tourist starts and ends at the same hotel every day.
- Constraints (4) and (5) state the relationship between x_{ijk}, y_i, and z_i.
- Constraint (6) ensures that only one hotel is selected for the entire tour.
- Constraints (7)-(12) are the time constraints.
- Constraint (13) is the budget constraint.

Note that the above MILP can be decomposed into m smaller sub-problems, each with only one hotel selection. It is clear that the sub-problems are much easier to solve, however, it is still very difficult to solve each individual sub-problem.

We note that TPP is related to the classical Vehicle Routing Problem (VRP). In VRP, the objective is to design the routes for a fleet of vehicles to service all customers with minimum total cost, subject to vehicle capacity and time constraints. The cost in VRP can be either the number of vehicles used or the total length of the routes. Graphically, VRP is to find a set of cycles with minimum total cost to cover all the vertices. Clearly, the Hamilton cycle problem is a special case of VRP. By comparison, TPP is to generate a fixed number of cycles such that the total utility value of the vertices covered by these cycles is maximized. Since VRP is a well-known NP-complete problem, it is not practical to obtain optimal solutions, even for medium-size problems; hence heuristics and approximate algorithms are often used to obtain near optimal solutions (e.g., see [9], [10], [11], [12]). TPP is also a NP-complete problem, thus in [13] we proposed an efficient heuristic method based on local search ideas. The method is quite efficient, and it can generate good solutions in relative short time. In what follows, we outline the basic steps involved in the method:

Step 1: Generate an initial solution by using some greedy algorithm.

Step 2: Select *p* sites at random in the current solution. For each site selected, replace it with a new site based on some selection rules (various selection rules can be developed, and they can be probabilistic) such that the new solution is feasible. The best solution among these *p* solutions is selected as the next solution if it is better than the current solution. This step is repeated until a desired solution is found or after a number of iterations.

There is significant feasibility in selection rules used in Step 2. In [13], we developed a local search algorithm for the selection of the new site. The numerical experiments presented in the following section are obtained based this local search algorithm, which seems to work quite well.

3. Numerical Experiments

In this section, we provide some numerical experiments from two examples (one small and one large). Both examples were taken from our work in a real-world problem. Example 1 has 30 sites and 3 hotels, with three different utility functions. We assume that there is no constraint on the total budget as well as the time during which the tourist can visit a site. We also assume that T_k is the same for all d days, so let $T_k = T$. We used CPLEX to solve the MILP and compared the results with those obtained based on our method. Our selection rules are probabilistic, so for our method we ran 30 replicates for each instance, based on which the average, maximum, and minimum objective values were then obtained. All our numerical experiments were run on a Pentium(R) 4 processor 1.7 G PC. The numerical results for Example 1 are presented in Tables 1 — 3. Based on these results it is clear that our method works extremely well: in 19 cases the error of our method are almost zero while in other 8 cases the error is less than 1%, where the error = (Optimal value – Average value of our method)/Optimal value. The average CPU times of our method are between 0.083 and 0.7 second, which are considerably less than those of CPLEX.

Example 2 has 59 sites and 26 hotels and is otherwise similar to Example 1. For this example, it is no longer possible to use CPLEX to solve, so only the heuristic method was applied. The numerical results are presented in Table 4. It is clear that our method produces reasonably good solutions in fairly short amount of times (in fact, all within 10 seconds.

4. Conclusion

In this paper, we studied the tour planning problem. We formulated it as a mixed-integer linear program problem, and proposed a heuristic method based on the idea of local search. We provided some numerical experiments that demonstrate that the method is quite efficient and is able to find good approximate solutions in very short time.

References

[1] *World Travel & Tourism Council, http://www.wttc.org/.*

[2] *D. Buhalis, eTourism: Information Technology for Strategic Tourism Management, Prentice Hall, 2003.*

[3] *BurstMeia Online Insights, Vol.2, 2007.*

[4] *W. Hannes, "Intelligent System in Travel and Tourism", International Joint Conference on artificial intelligence, 2003.*

[5] *D. Buhailis, R. Law, "Twenty Years on and 10 Years after the Internet: the Sstate of eTourism Research", Tourism Management 2008, Vol. 29, No. 4, pp.609-623, 2008.*

[6] *J.B. Schafer, J. Konstan, J. Riedi, "Recommender Systems in e-commerce", Proceedings of the 1st ACM conference on Electronic commerce, 1999.*

[7] *F. Ricci, "Travel Recommender Systems", IEEE Intelligent Systems, Vol. 17, No.6, pp.55-57, 2002.*

[8] *F. Ricci, H. Wether, "Case Base Querying for Travel Planning Recommendation", Information Technology & Tourism, Vol. 4, No.3-4, pp. 215–226, 2002.*

[9] *B. Olli and G. Michel, "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms", Transportation Science, Vol.39, No.1, pp.104-118, 2005.*

[10] *B. Olli and G. Michel, "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics", Transportation Science, Vol.39, No.1, pp. 119-139, 2005.*

[11] *P. David and R. Stefan, "A General Heuristic for Vehicle Routing Problems", Computers & Operations Research, Vol.34, No.8, pp.2403-2435, 2007.*

[12] *M. Gendreau and J. Y. Potvin, "Metaheuristics in Combinatorial Optimization", Annals of Operations Research, Vol.140, No.1, pp. 189–214, 2005.*

[13] *Chenbo Zhu. "The Tour Planning Problems", Technical Report, School of Management, Fudan University, 2009.*

Table 1: Example 1 with Utility Function 1

d	T	Our Method				MILP		Error (%)
		Objective Value			Average CPU Time (seconds)	Objective Value	CPU Time (senconds)	
		Minimum	Average	Maximum				
1	8	239	239	239	0.087	239	0.438	0.0000
	10	314	314	314	0.105	314	0.828	0.0000
	12	344	344	344	0.343	344	2.109	0.0000
2	8	422	422	422	0.308	422	1.844	0.0000
	10	529	529	529	0.336	529	54.141	0.0000
	12	634	634	634	0.582	634	440.547	0.0000
3	8	576	576	576	0.388	576	12.219	0.0000
4	8	727	727	727	0.590	727	29.422	0.0000
5	8	865	865	865	0.670	865	78.359	0.0000

Table 2: Example 1 with Utility Function 2

d	T	Our Method				MILP		Error (%)
		Objective Value			Average CPU Time (seconds)	Objective Value	CPU Time (senconds)	
		Minimum	Average	Maximum				
1	8	278	278	278	0.084	278	0.625	0.0000
	10	370	370	370	0.095	370	0.703	0.0000
	12	440	449.8	454	0.373	454	1.625	0.9251
2	8	508	508	508	0.311	508	3.906	0.0000
	10	656	669.4	675	0.429	675	31.937	0.8247
	12	792	800.2	808	0.603	808	918.563	0.9612
3	8	714	716.3	717	0.457	717	22.141	0.0976
4	8	908	913.6	914	0.611	914	175.078	0.0438
5	8	1108	1108	1108	0.689	1108	1388.748	0.0000

Table 3: Example 1 with Utility Function 3

d	T	Our Method				MILP		Error (%)
		Objective Value			Average CPU Time (seconds)	Objective Value	CPU Time (senconds)	
		Minimum	Average	Maximum				
1	8	187	187	187	0.083	187	0.422	0.0000
	10	248	248	248	0.083	248	1.188	0.0000
	12	284	284	284	0.341	284	2.187	0.0000
2	8	332	343.0	345	0.330	345	2.563	0.5700
	10	437	439.4	443	0.378	443	68.047	0.8126
	12	525	527.8	529	0.535	529	1416.908	0.2268
3	8	501	501	501	0.438	501	10.594	0.0000
4	8	642	642	642	0.658	642	35.531	0.0000
5	8	776	776	776	0.700	776	341.469	0.0000

Table 4: Example 2

d	T	Utility Function 1				Utility Function 2				Utility Function 3			
		Objective Value			Avg. CPU Time (secs)	Objective Value			Avg. CPU Time (secs)	Objective Value			Avg. CPU Time (secs)
		Min.	Avg.	Max.		Min.	Avg.	Max.		Min.	Avg.	Max.	
1	8	259	259	259	1.033	287	287	287	1.073	187	187	187	1.537
	10	314	314.8	315	0.919	370	370	370	1.817	248	248	248	1.978
	12	370	370	370	0.916	432	449.7	454	0.988	283	288.6	291	0.969
2	8	419	428.0	429	3.193	505	510.3	516	1.894	330	342.2	345	1.469
	10	523	542.2	554	1.834	650	665.4	675	2.098	423	444.9	456	3.034
	12	632	644.5	663	2.438	770	788.4	808	2.000	529	537.0	544	1.930
3	8	598	602.7	612	3.057	705	743.7	745	1.969	471	495.2	501	1.936
	10	726	754.4	767	4.831	898	931.8	948	3.306	608	624.9	650	3.705
	12	880	904.6	934	4.047	1086	1109.1	1136	4.047	749	767.5	777	3.071
4	8	740	760.7	778	4.290	873	896.6	923	3.263	592	605.5	629	1.709
	10	920	949.3	976	3.674	1137	1166.5	1190	3.389	763	789.1	807	2.814
	12	1121	1146.4	1157	4.448	1380	1402.2	1432	3.689	934	956.7	991	2.904
5	8	904	915.5	933	6.684	1060	1095.4	1136	5.448	712	727.1	754	2.518
	10	1116	1140.1	1159	1.140	1347	1390.4	1416	4.270	904	941.0	980	3.920
	12	1357	1379.7	1421	7.051	1653	1677.8	1709	4.192	1101	1123.9	1153	3.891
6	8	1010	1054.6	1073	9.153	1231	1272.9	1293	8.209	813	838.9	891	3.126
	10	1292	1319.4	1345	6.224	1575	1603.5	1631	5.619	1034	1075.4	1107	4.576
	12	1564	1589.6	1618	8.661	1890	1927.3	1970	8.229	1240	1267.4	1296	5.721