Periodica Polytechnica
Transportation Engineering

45(3), pp. 133-140, 2017
https://doi.org/10.3311/PPtr.9916

RESEARCH ARTICLE

Practical Route Planning Algorithm

Sándor Apáthy M.[1*]

Received 21 August 2016; accepted 08 December 2016

Abstract
Routing algorithms are traditionally considered to apply the sum of profits gathered at visited locations as an objective function since the Traveling Salesman Problem. This heritage disregards many practical considerations, hence the result of these models meet with user's needs rarely.
Thus considering the importance of this theoretical and modeling problem, a novel objective function will be presented in this paper as an extension of the one inherited from the TSP that is more aligned with user preferences and aims to maximize the tourist's satisfaction. We also propose a heuristic algorithm to solve the Team Orienteering Problem with relatively low computation time in case of high number of vertices on the graph and multiple tour days. Based on the key performance indicators and user feedback the algorithm is suitable to be implemented in a GIS application considering that even a 3-day tour is designed less than 4 seconds.

Keywords
Team Orienteering Problem, Route Planning, Heuristic Algorithm, Tourism
JEL code: C60, C61, Z32

[1] Department of Mathermatics,
Corvinus University of Budapest
H-1093 Budapest, Fővám tér 13-15, Hungary
[*] Corresponding author, e-mail: sandor.m.apathy@gmail.com

1 Introduction

Routing algorithms are traditionally considered to apply the sum of profits gathered at visited locations as an objective function since the Traveling Salesman Problem. This heritage disregards many practical considerations, hence the result of these models meet with user's needs rarely. In this paper after briefly introducing the related research activities, a novel heuristic routing algorithm is to be presented, that aims to provide personalized tour plans for the users. Thus a novel objective function will be presented in this paper as an extension of the one inherited from the TSP that is more aligned with user preferences and aims to maximize the tourist's satisfaction. Then in section 4 we propose a heuristic routing algorithm to provide near optimal solution for the previously defined Team Orienteering Problem with relatively low computation time even in case of high number of vertices on the graph and multiple tour days. Based on the experiences, presented in section 5, the algorithm is suitable to be implemented in a GIS application considering that even a 3-day tour is designed less than 4 seconds. The performed user evaluation test confirmed that the proposed objective function provides significantly better route plans than the conventional approach inherited from the Traveling Salesman Problem. It is also important to highlight that as of our knowledge there is no such existing solution, that is able to deliver route plan for multiple days, assigned and optimised to a dedicated hotel. It is also important to highlight that similar problem occurs in several fields and it is widely researched, though as of our knowledge there is no such existing Touristic solution, that is able to deliver route plan for multiple days, assigned and optimised to a dedicated hotel.

2 Related work

Probably the first milestone on the field of route planning was the *Traveling Salesman Problem* (TSP) formalized by Karl Menger (1928), though it was named by Hassler Whitney (Schrijver, 2005). The agent knows the travel cost of each arc and he has to minimize the overall travel cost while visiting all vertices. Thanks to Birkhoff's (1946) work assignment problems can be solved by the *simplex method*, which has been used first by Danzig et al. (1954) to solve TSP.

Orienteering Problem (OP) or often called *Selective Travelling Salesman Problem* unfolded from the TSP, where each vertex has a profit assigned and each arc has a travel time cost. The agent has to visit vertices on the graph to maximize his profit while keeping the time boundary. The appellation originates from Chao et al. (1996a), though the problem was first formulated years before by Tsiligirides (1984). He applies stochastic algorithm to provide an approximation of the optimum, and searches the next vertex in each iteration by *Monte Carlo method*. OP was also formalized by Kataoka and Morito (1988), albeit they labeled it as *Maximum Collection Problem*. Feillet et al. (2004) discuss the topic in more detail in their survey. OP appellation originates from the popular orienteering sports where one needs navigational skills and stamina to visit all checkpoints in the shortest time. It was the first area where the discipline of route planning solved a non-industrial problem, then it unfolded in other areas, such as hiking, biking and city trip planning. Wang et al (2008) are using OP techniques to design 1 day trips in a city starting and ending the tours in a given hotel. Golden et al. (1984) have proven that the OP is NP-hard, so exact solutions can only be given in case of small number of peaks. Ramesh et al. (1992) have applied B*ranch-and-bound algorithm* to provide exact solution for the OP on a graph of 150 vertices, while Fischetti et al. (1998) have been able to obtain exact solution with similar method even up to 500 vertices in relatively short computational time. *Opt-2* and *opt-3* methods have been applied in 4 phases by Ramesh and Brown (1991) to give a heuristic solution for the same problem. Chao et al. (1996b) outperform these results by incorporating a greedy and an *opt-2* algorithm with stochastic methods in 5 steps to define a near-optimal route. In some case these heuristic methods can stuck in a local-optimum point that is effectively eliminated by tabu search algorithms, e.g. Gendreau et al. (1998). As application of these results are recently gained focus, growing number of papers are concerning with the GIS implementations, particularly in mobile applications, e.g. Souffriau et al. (2008).

Team Orienteering Problem (TOP) is a natural extension of OP, where the agent has P days to visit vertices on the graph maximizing the sum of profits within the given time boundaries, while each route starts and ends in a given hotel. TOP was first formalized by Butt and Chavalier (1994) to solve a recruitment problem. *Column Generating algorithms* perform effectively, where the problem is redefined and solved as an LP problem, though dimension is reduced to obtain better processing time, as we can see by Butt and Ryan's (1999) exact solution for 100 vertices in a relatively short computational time. Boussier et al. (2007) have reached better performance by incorporating column generation with a branch-and-bound method. Tabu search algorithm has been used to solve TOP by Tang and Miller-Hooks (2007) and also by Archetti et al. (2007). Ke et al. (2008) applied Ant Colony algorithm to provide a heuristic solution. As a first step 4 method have been tested simultaneously to obtain a viable solution, where they found the sequential algorithm to be the most effective. Then in each iteration they improve the solution with an opt-2 algorithm and supplement the routes with additional vertices until reaching the boundaries. Vansteenwegen et al. (20089a) proposed two heuristic solutions, the *Guided Local Search* (GLS) and the *Skewed Variable Neighborhood Search* (SVNS), Vansteenwegen et al. (2009b), consisting of the same steps: after obtaining an initial solution the "less profitable" route sections are eliminated, then smaller routes are joined and optimized by swapping and replacing. SVNS clearly outperforms GLS by applying a different sequence of these steps. Further readings can be found in Vansteenwegen et al. (2011b), where computation times are also highlighted in each case.

3 Problem formalisation

We intend to solve the Team orienteering problem, formalized first by Butt and Cavalier (1994), though as we do not necessarily agree with the maximization of sum of profit points by visiting each arc, we recommend a new approach. This objective function has been inherited from the Traveling Salesman Problem, formalized and solved first by Karl Menger in the late 20s, see Ramesh et al. (1992), where we can attribute particular meaning to collecting profit points by visiting the vertices. As per my understanding tourists are not interested in attraction they have evaluated poorly (namely assigning 4 or less points on a scale 1 to 10). Hence we decided to design an objective function as an extension of the broadly applied version. Let the utility function be $u(s_i,a)$, where s_i is the evaluation of the vertex *i* by a certain user, and *a* is a parameter that measures how much the user appreciates an attraction evaluated as *s* comparing to *s-1*. The utility function is designed as follows:

$$u(s_i,a)=\begin{cases}\dfrac{1-e^{-a\left(s_i-s^*\right)}}{a}\Big|a\neq 0\\ s_i-s^*\,\big|a=0\end{cases}$$

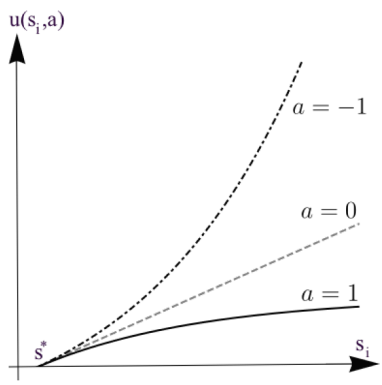


Fig. 1 Utility function

The effect of parameter a on the evaluation is depicted on Fig. 1. By using the utility function determined above we have the opportunity to rule out POIs that are undesirable for the user (namely with evaluation under s^*). Thus we avoid visiting POIs that are very close to our planned route, despite their poor evaluation.

We use the following notation in the problem formalization:

- N - number of vertices in graph $G(N,E)$
- E - number of arcs in graph $G(N,E)$
- P - number of available days
- B - daily budget constraint (the user only have to meet with the overall budget constraint BP)
- T_{max} - daily time limit (the user has to meet with the daly time limit every day, and he can only exceed it with 5%)
- s_i - evaluation of vertex i
- v_i - visit time of vertex i
- t_{ij} - travel time from vertex i to j
- τ_{ijp} - is equal to 1 if in the pth path the vertex j is visited right after vertex i, 0 in any other cases
- θ_{ip} - is equal to 1 if in the pth path the vertex i is visited, 0 in any other cases
- β - parameter of "laziness"
- α - parameter of the objective function to balance the importance of utility and visiting time - travel time ratio
- R - denotes the set of planned routes for P days
- b_i - entrance fee for vertex i

The objective is to design P routes for P days that maximizes the objective function and observe the constraints. All vertices of the graph can be visited not more than once, except the hotel. Each route starts at the hotel and ends at the hotel (vertex number 1 and N denotes the hotel). The problem is formulated as follows:

$$\max_{\tau_{ijp}} \left(\frac{\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} u_i}{\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij}^{\beta}} \right)^{\alpha} \times \left(\sum_{p=1}^{P} \sum_{n=1}^{N} \theta_{ip} u(s_i, a) \right)^{1-\alpha}$$

$$\sum_{p=1}^{P} \sum_{j=2}^{N} \tau_{1jp} = \sum_{p=1}^{P} \sum_{i=1}^{N-1} \tau_{iNp} = P$$

$$\sum_{p=1}^{P} \theta_{kp} \leqslant 1; \forall k = 2, \ldots, N-1$$

$$\sum_{j=2}^{N} \tau_{kjp} = \sum_{i=1}^{N-1} \tau_{ikp} = \theta_{kp}; \forall k = 2, \ldots, N-1; \forall p = 1, \ldots, P$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij} + \sum_{i=1}^{N} \theta_{ip} v_i \leqslant T_{\max}; \forall p = 1, \ldots, P$$

$$\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} b_i \leq PB$$

$$h_{ip} - h_{jp} + 1 \leqslant (N-1)(1-\tau_{ijp}); \ \forall i, j = 2, \ldots, N; \forall p = 1, \ldots, P$$

$$2 \leqslant h_{ip} \leqslant N; \forall i = 2, \ldots, N; \forall p = 1, \ldots, P$$

$$\tau_{ijp}, \theta_{ip} \in \{0,1\} \forall i, j = 1, \ldots, N; \forall p = 1, \ldots, P$$

The interpretation of the lines:

1. Objective function to be optimized, where the two considered factors are the sum of visit times divided by the sum of travel times and the sum of utilities earned by visiting vertices.
2. Every route starts at vertex 1 and ends at vertex N (both denotes the same hotel).
3. Every vertex is visited once at most.
4. Every route is connected severally.
5. The routes are meeting with the time constraint for each day.
6. The routes are meeting with the budget constraint for the overall tour of P days.
7. and 8. together grants avoiding cycles in the route according to Miller et al. (1960).
9. The value set of τ_{ijp} and θ_{ip} is 1 or 0.

Where our problem differs from the literature is the objective function which is clearly an extension of the usual profit collecting (since it yields the same formula for $\alpha = a = 0$). According to the objective function there are main goals to complete: maximize the tourist's sum of utilities and keeping low the visiting time and travel time ratio. Parameter α enables us to calibrate the weight of these objectives and keep balance.

4 Routing Algorithm

4.1 Functions

As the Orienteering Problem is NP-hard, hereby we intend to provide a novel heuristic algorithm to find a near-optimal solution for the problem defined in the previous section. We start presenting our routing algorithm with two functions:

Lexicographical ranking: We determine the rank of a set of vertices (C_1) to another set of vertices (C_2) considering the effectiveness of the constraints and the score threshold. More formally $L(C_1, C_2, sc, s^*)$, where sc denotes the resource that is more scarce (the estimation is going to be presented in the 3rd step of the algorithm) and C_1 is the set of vertices to be ranked based on the set of vertices in C_2. Let us calculate the two measures below:

$$\frac{\dfrac{u(s_i, aj)}{u(s^*+1, a)}}{d^*(c_i, C_2) + v_i} \qquad \frac{\dfrac{u(s_i, a)}{u(s^*+1, a)}}{b_i}$$

where $d^*(c_i, C_2)$ denotes the mean distance between vertex c_i (element of C_1) and the elements of C_2. The nominator interprets the utility of s_i in the units of the utility of s^*+1 (which is the lowest score assigned to a vertex we consider to be visited, see the first step of the algorithm). In case of the second measure we divide the utilities by the corresponding entrance fee. $L(C_1, C_2, sc, s^*)$ first determines the scarce constraint, e.g. let it be the time. In this case it ranks the vertices of C_1 based on the first measure, cuts the list into 6 equal pieces (the number of elements in the last group can differ), then it rerank the vertices within each group based on the second measure.

Identifying outliers: *O(H, cr)* determines the outliers for a given Hamiltonian-cycle considering the *cr* threshold parameter. First we calculate for each vertex of the Hamiltonian-cycle the sum of travel times of the inbound ($t_{i,in}$) and outbound ($t_{i,out}$) arcs. Vertex i is considered to be an outlier in case:

$$t_{i,in} + t_{i,out} > t_H^* + cr\sigma_H$$

where t_H* is the mean of inbound and outbound travel time and σ_H is the standard deviation. The value of *cr* differs in some cases that we indicate later.

4.2 The algorithm

The algorithm consists of the following steps:

1. Simplification of the problem space: let us eliminate all vertices with evaluation less than or equal to *s**. We call the remaining vertices as the set of relevant points, denoted by C_r
2. Fixed vertices: let us appoint mandatory vertices, practically vertices with maximum evaluation ($\max_i\{s_i\}$).
3. Grouping: Let us first estimate which is the more scarce resource. As a rule of thumb we calculate the below two measures:
4. Take the sum of visiting times of relevant points and the average distance between them (once for each) and divide it by the overall time constraint PT_{max}
5. The sum of entry fees for all relevant points divided by the budget constraint *BP*
6. The higher measure is considered to appoint the more scarce resource (so *sc* is determined). Let us choose the vertices with the highest score (as those are mandatory) then for all the remaining ones use $L(C_1, C_1, sc, s^*)$ and pick the first 5*P* points and add them to the mandatory vertices. Now determine an optimal Hamiltonian-cycle (including the hotel) on the vertex set, then apply *O(H,1)* to eliminate outliers. As it was stated earlier no fixed vertex can be considered as an outlier.
7. Initial daily routes: The remaining vertices are to be separated into P groups (without any overlap) by Hartigan-Wong-algorithm, see Hartigan - Wong (1979). The computational cost of this method is dominated by the complexity of sorting, which is O(N logN). For each cluster we determine the shortest Hamiltonian-cycle (including the hotel), and choose the best result based on 10 iterations. We consider a solution to be better in case the objective function's value is higher for the *P* routes altogether. It is plausible that maximizing the objective function on a graph of fixed vertices it is equivalent with finding the shortest Hamiltonian-cycle on the graph (where the travel costs are on the βth power), as the visiting times and utilities are invariable in the objective function. Thus we used R software's *Repetitive Nearest Neighbor algorithm* to optimize TSP, Gutin et al. (2002).
8. Refill: In case of any free capacities in each day we refill the routes with the remaining points from C_r, hence we use $L(C_r, C_i, sc, s^*)$ for each day i, then fill the routes until we exceed any of the resource constraints by 20% (at this stage we allow violating the constraints). It is important to note that in each time a vertex is eliminated, it instantly relocated to set C_r. In case a vertex is inserted to a route of any day, it is eliminated from set C_r.
9. Switch: For each point in all existing routes the mean of the 3 lowest distances from the points of the regarding route are to be determined (we also calculate this measure for the route contains the given point), and we assign the point to the route where the value of this measure is the lowest, and repeat it for 10 iterations.
10. Cut: In case a day exceeds the time constraint with more than 5%, we apply $L(C_i, C_i, sc, s^*)$ and eliminate the vertices from the last in the rank until we meet with the constraint.
11. Refill: If we still have free capacities at day i, we apply $L(C_r, C_i, sc, s^*)$ and insert points from C_r from the top rank until we meet with any of the constraints.

5 Experimental results

We have performed our experiment on a POI list consists of 150 touristic attractions in Budapest, including the visit times, entry fees, locations (longitude and latitude values) and the evaluations given by the users (that is assumed to be calculated based on our recommender system). The travel times are derived from the location applying OpenStreetMap API to calculate the distance matrix for 150 POIs and the hotel assigned to the user. Hereby we summarize our experiences:

We obtain large bypasses in case of positive a parameter values, moreover the routes disintegrate when we apply low parameter values for α and β.

- In case $\alpha < 0.5$ acceptable routes can only be obtained if $\beta > 1.5$ and $a < -0.5$
- Generally relatively good results have been observed in case $a < -0.5$; $\alpha > 0.5$ and $\beta > 1.5$
- Considering the chargeability and the visiting time - travel time ratio as a measure of goodness we found $\alpha = 0.75$; $a = -1$ and $\beta = 2$ parameter-set optimal in case of 1–2–3 and 4 days, where chargeability denotes to what extent we utilized the available daily time limit (T_{max}). We present in *Appendix A* a 4-day tour calculated applying these parameter values, and as a point of comparison the result of the "profit collection" case ($\alpha = a = 0$) is also presented. The "profit collection" case has shown poor results: though the chargeability is almost equal to our solution (95% average) and the computation time is lower by 1 second, the visit time - travel time ratio is 1.43 comparing to 3.8 in case of our parameter values, that indicates spending relatively much more time at the

attractions than on traveling in the proposed solution comparing to the "profit collection" case.

- We ran the algorithm 100 times to calculate a 3 day tour, and we obtained 3.73 seconds mean computation time (we ran our tests in R software on a laptop with following details: 3.8 GB RAM, Intel Core i3-3217U CPU, 1.80GHz × 4 proc). Though we cannot compare the results to Vansteenwegen et al. (2011a) or Gavalas - Kenteris (2011), as we solved a different problem, as a point of comparison we mention the results of Sylejmani - Dika (2011), who presented a tour planner algorithm. The designed *Taboo Search algorithm* has created a tour of 3 days on a graph of 40 vertices with a mean computation time of 81.7 seconds.
- We also present the results of the test runs for each parameter combination (performing 20 iterations each). As we can see on Fig. 2. the mean computation time increasing significantly between the tours designed for 2 and 3 days. The average visit time - travel time ratio and the average chargeability slightly decreases by increasing the number of days.
- Based on the observation computation time is invariant to parameter a, while β slightly increases computation time in case of 3 or 4-day tours. Increasing a significantly increases computation time in case of multiple day tours. Most probably it is in relation with the fact that in these cases the algorithm struggles to find vertices that would increase the value of the objective function. In *Appendix B* we summarized how parameters are effecting chargeability, computation time and visit time - travel time ratio for 1–4-day tour cases.
- Visit time - travel time ratio (P-day tour overall) shows decreasing tendency by increasing the number of tour days, as the additional vertices are often located at a larger distance. While parameter a and β does not effect the ratio, increasing parameter α significantly increases it.
- In terms of approximability it is need to be highlighted that the traveling salesperson problem is strictly complete for the class of NP minimization problems w.r.t. any cost-respecting quality measure µ (Orponen and Mannila, 1990).

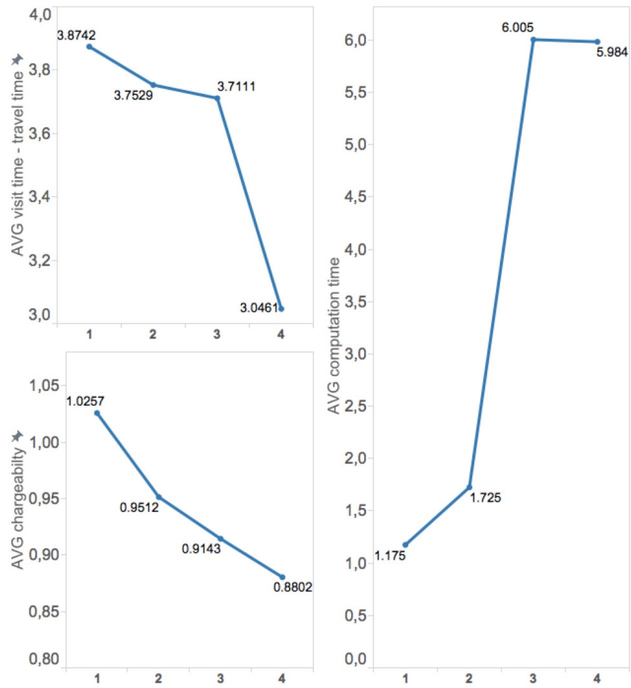


Fig. 2 Test results of the routing algorithm

6 User evaluation

A user evaluation test has been performed to compare the proposed objective function to the broadly applied approach inherited from the Traveling Salesman Problem that maximises the sum of profit points of visited arcs. The test application designs route plans for 1 to 4 days in 4 European cities (Budapest, London, Paris and Rome) after the users proved information on their travel preferences: daily budget (time and money), willingness to walk (3 options), number of days spent in the city, accommodation (6 options in each city) and POI preferences (17 types to be evaluated on a 4-level scale, e.g. museums, religious place, park and nature, etc.). Then the users have the opportunity to rate the proposed two route plans on a 10-level scale (where 1 denotes the worse recommendation the user can imagine, while 10 represents a near perfect recommendation).

Based on the web survey populated by 67 users (*Fig. 3.*) the proposed objective function received significantly higher average user evaluation (8.16) than the conventional approach (6.448), as the performed t-test confirmed the results (see *Appendix C*). As we consider user satisfaction to be the ultimate measure of success in tour plan design, the test results reinforced our belief in the new approach and determine the future research direction.

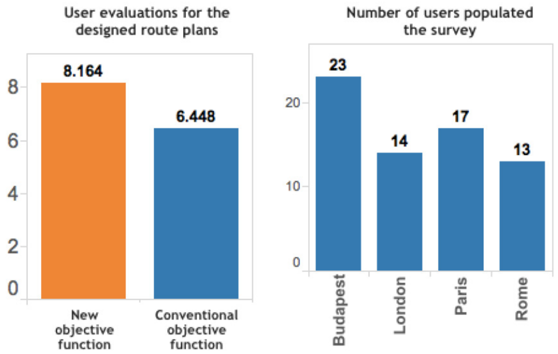


Fig. 3 Test results of the user evaluation

7 Conclusions and future work

A modified version of the TOP has been presented, that - at least according to our intentions - focusing more on a practical problem formulation related to the Touristic trip design problem. As part of our endeavor we designed an objective function that adjust the problem formulation according to the user's preferences. A novel heuristic solution has been introduced to solve the TOP which provides an opportunity implementing in a mobile tour planning application considering its low computation time. Based on the experimental results the proposed solution outperforms the "profit collection" approaches, though as these considerations are clearly subjective. As we aim to serve tourist's needs, our approach has been tested on a larger set of users and resulted in significantly better average evaluation comparing to "profit collection".

As a future goal the algorithm is to be extended with the ability to handle time windows which is considered to be the next milestone towards a practically useful solution. By looking over more hotels and appointing the one where the value of the objective function reaches its maximum for the P days tour, the present algorithm would be able to satisfy further touristic needs. Moreover the first clustering step with a method that identifies P vertices in sufficient distance from each other and builds trees including the hotel is to be replaced with a more effective method to further decrease computation time.

References

Archetti, C., Hertz, A., Speranza, M. (2007). Metaheuristics for the team orienteering problem. *Journal of Heuristics*. 13, pp. 49-76. https://doi.org/10.1007/s10732-006-9004-0

Birkhoff, G. (1946). Tres observaciones sobre el algebra lineal, Revista Facultad de Ciencias Exactas, Puras y Aplicadas Universidad Nacional de Tucuman, Serie A (Matematicas y Fisica Teorica), 5, pp. 147-151. (in Spanish)

Boussier, S., Feillet, D., Gendreau, M. (2007). An exact algorithm for the team orienteering problem. *4OR*., 5, pp. 211-230. https://doi.org/10.1007/s10288-006-0009-1

Butt, S. E., Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*. 21, pp. 101-111. https://doi.org/10.1016/0305-0548(94)90065-5

Butt, S., Ryan, D. (1999). An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & Operations Research*. 26, pp. 427-441. https://doi.org/10.1016/S0305-0548(98)00071-9

Chao, I., Golden, B., Wasil, E. (1996a). Theory and methodology – a fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*. 88, pp.475-489.

Chao, I., Golden, B., Wasil, E. (1996b). Theory and methodology – the team orienteering problem. *European Journal of Operational Research*. 88, pp. 464-474.

Dantzig, G., Fulkerson, R., Johnson, S. (1954). Solution of a Large Scale Traveling Salesman Problem. *Journal of the Operations Research Society of America*. 2, pp. 393–410.

Feillet, D., Dejax, P., Gandreau, M. (2004). Traveling Salesman Problems with Profits. *Journal of Transportation Science*. 39(2), pp. 188-205.

Fischetti, M., Salazar, J., Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*. 10(2), pp. 133-148. https://doi.org/10.1287/ijoc.10.2.133

Gavalas, D., Kenteris, M. (2011). A web-based pervasive recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing*. 15(7), pp. 759–770. https://doi.org/10.1007/s00779-011-0389-x

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., Vathis, N. (2015). Heuristics for the Time Dependent Team Orienteering Problem: Application to Tourist Route Planning. *Computers & Operations Research*. 62, pp. 36-50. https://doi.org/10.1016/j.cor.2015.03.016

Gendreau, M., Laporte, G., Semet, F. (1998). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*. 106(2-3), pp. 539-545. https://doi.org/10.1016/S0377-2217(97)00289-0

Golden, B., Levy, L., Vohra, R. (1984). The Team Orienteering Problem. *Naval Research Logistics Quarterly*. 34, pp. 307–318.

Gutin, G., Yeo, A., Zverovich, A. (2002). Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*. 117(1-3), pp. 81-86. https://doi.org/10.1016/S0166-218X(01)00195-0

Hartigan, J. A., Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C*. 28(1), pp. 100-108.

Kataoka, S., Morito, S. (1988). An algorithm for the single constraint maximum collection problem. *Journal of the Operations Research Society of Japan*. 31(4), pp. 515-530.

Ke, L., Archetti, C., Feng, Z. (2008). Ants can solve the team orienteering problem. *Computers and Industrial Engineering*. 54, pp. 648-665. https://doi.org/10.1016/j.cie.2007.10.001

Menger, K. (1928). Ein Theorem über die Bogenlange, Anzeiger — Akademie der Wissenschaften in Wien — Mathematisch-naturwissenschaftliche, Klasse 65, pp. 264–266. (in German)

Miller, C., Tucker, A., Zemlin, R. (1960). Integer programming formulations and travelling salesman problems. *Journal of the ACM*. 7, pp. 326–329. https://doi.org/10.1145/321043.321046

Orponen, P., Mannila, H. (1987). On approximation preserving reductions: Complete problems and robust measures. Technical Report C-1987-28, Department of Computer Science, University of Helsinki

Ramesh, R., Brown, K. (1991). An efficient four-phase heuristic for the generalized orienteering problem. *Computers & Operations Research*. 18(2), pp. 151-165. http://dx.doi.org/10.1016/0305-0548(91)90086-7

Ramesh, R., Yoon, Y., Karwan, M. (1992). An optimal algorithm for the orienteering tour problem. ORSA Journal on Computing. 4, pp. 155-165. https://doi.org/10.1287/ijoc.4.2.155

Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). In: Handbook of Discrete Optimization. (Aardal, K., Nemhauser, G. L., Weismantel, R. (eds.)), Elsevier, Amsterdam, pp. 1–68.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., Van Oudheusden, D. (2008). A personalised tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*. 22(10), pp. 964-985. https://doi.org/10.1080/08839510802379626

Sylejmani, K., Dika, A. (2011). Solving touristic trip planning problem by using taboo search approach. *International Journal of Computer Science Issues*. 8(3), pp. 139-149.

Tang, H., Miller-Hooks, E. (2005). A tabu search heuristic for the team orienteering problem. *Computer & Operations Research*. 32, pp. 1379-1407. https://doi.org/10.1016/j.cor.2003.11.008

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*. 35(9), pp. 797-809. https://doi.org/10.2307/2582629

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D. (2009a). A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research*. 196(1), pp. 118-127. https://doi.org/10.1016/j.ejor.2008.02.037

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D. (2009b). Metaheuristics for tourist trip planning. In: *Metaheuristics in the Service Industry*, *Lecture Notes in Economics and Mathematical Systems*, (Geiger, M., Habenicht, W., Sevaux, M., Sörensen, K. (eds.)), Vol. 624, Springer Berlin Heidelberg, pp. 15–31. https://doi.org/10.1007/978-3-642-00939-6_2

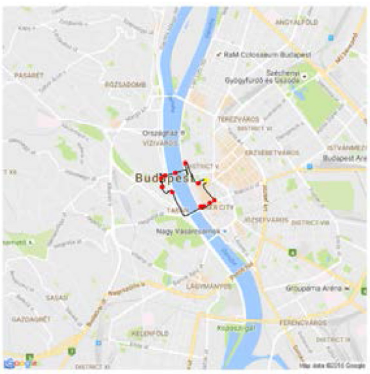
Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D. (2011a). The city trip planner: an expert system for tourists. Expert Systems with Applications. 38(6), pp. 6540–6546. https://doi.org/10.1016/j.eswa.2010.11.085

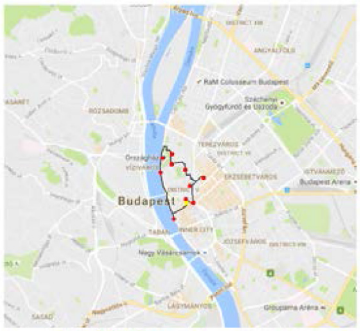
Vansteenwegen, P., Souffriau, W., Van Oudheusden, D. (2011b). The orienteering problem: a survey. *European Journal of Operational Research*. 209(1), pp. 1–10. https://doi.org/10.1016/j.ejor.2010.03.045

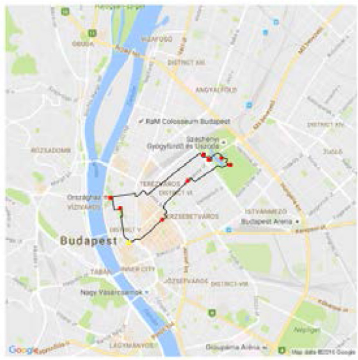
Wang, X., Golden, B., Wasil, E. (2008). Using a genetic algorithm to solve the generalized orienteering problem. In: The Vehicle Routing Problem: Latest Advances and New Challenges. (Golden, B., Raghavan, S., Wasil, E. (eds.)), Vol. 43, Springer US, pp. 263-274. https://doi.org/10.1007/978-0-387-77778-8_12

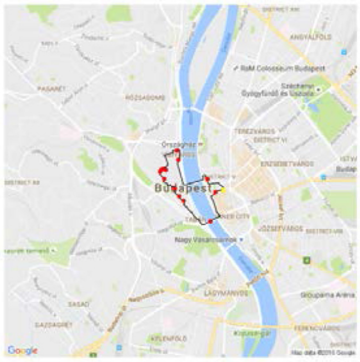
Appendix A

3 day tour (α = 0.75; a = –1 and β = 2)

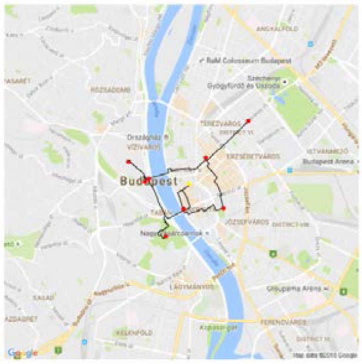


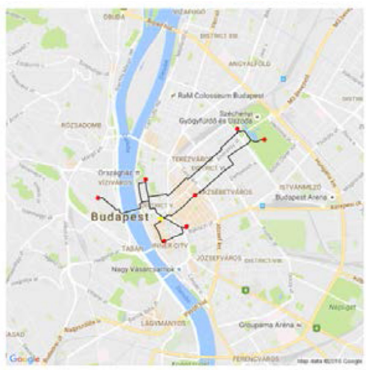


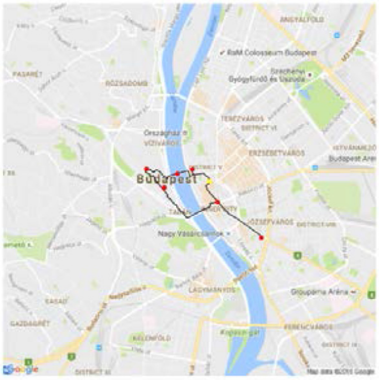


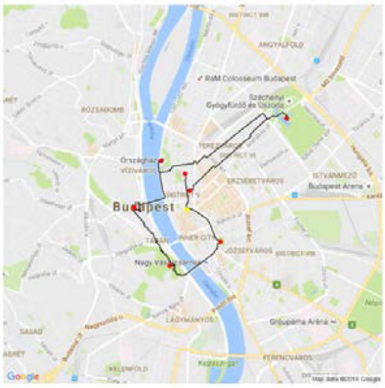


3 day tour (α = 0; a = 0)



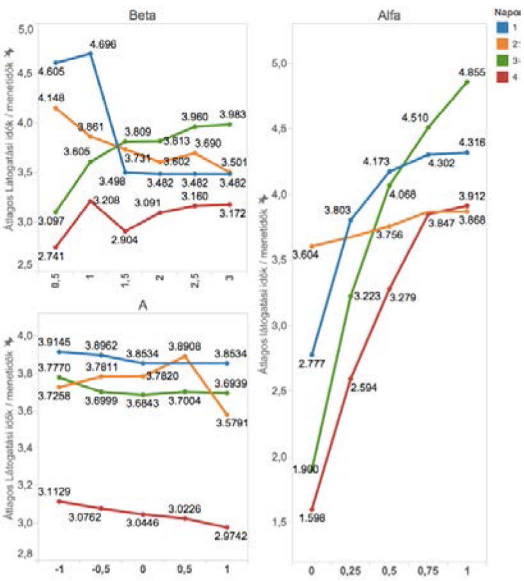


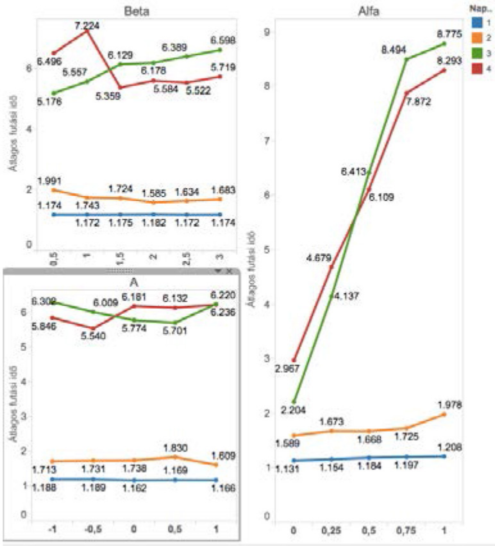


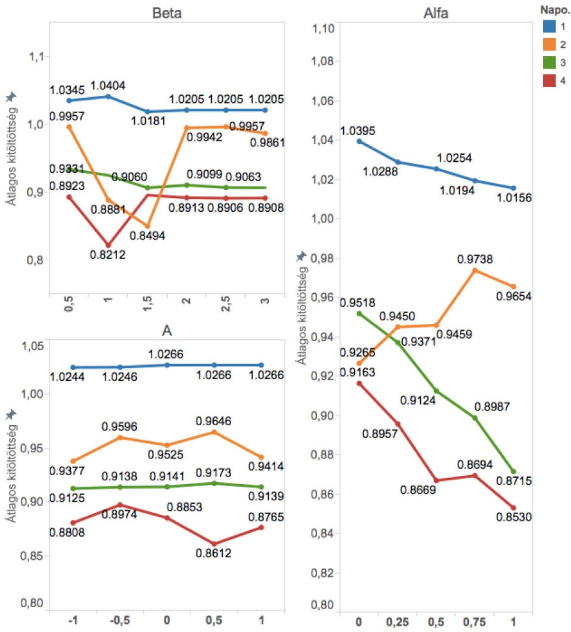


Appendix B

Results of the routing algorithm considering parameter values.







Appendix C

Results of the Welch-test comparing the two objective functions

	test values
t-value	7,1172
p-value	1.745e-11
avg of new approach	8,1810
avg of conventional approach	6,3879
DF	208,44
95% conf int.	1,2964
95% conf int.	2,2898
Ho	rejected