Automatisierte Wanderroutenerstellung als Maximierungsproblem - Zusammenfassung Literaturrecherche

May 31, 2022

1 Orienteering Problem

The orienteering problem (OP) is defined like this: Let $N = \{n_1, ..., n_{|N|}\}$ be a set of nodes in a contiguous graph such that each node $n_i \in N$ has a score s_{n_i}. The goal of the OP is to visit a set of nodes $N^* \subseteq N$ so that the score among the visited nodes is maximized and a given time limit is not violated.

The OP very closely matches the basic goal of our project to build a web application that generates the most interesting/best walking route among a subset of POIs around the chosen start and end points without violating time constraints, which is why we focused on the different extensions and applications of the OP for our further research.

There are plenty of research articles that investigate the OP and its variants/extensions, respectively; these will be further discussed in the following sections.

2 Generalized Orienteering Problem

The Generalized Orienteering Problem (GOP) [6, 8] differs from the classical OP by its scoring. Instead of having a single score s_{n_i}, all nodes are assigned m attribute scores $s_{n_i,1},\ s_{n_i,2},\ s_{n_i,m}$. Then, a weight function with $\sum_{x=1}^{m} W_x = 1$ is provided which assigns each of the attributes a weight, to calculate a total score. This means that the total score of a path is calculated by $S_n = \sum_{x=1}^{m} W_x [\sum_{i \in N} \{s_{n_i,x}^k\}]^{1/k}$. Parameter k allows for adjustment whether weighting is applied in a linear fashion. Two algorithms are proposed to solve the GOP.

One of them is a genetic algorithm. This means that a solution is found by a process that works similar to biological evolution. The pool of currently considered solutions is called a population. The population consists of chromosomes, which in turn consist of many genes. A gene in a genetic algorithm is partial information about a solution, so the genes in a chromosome together, form one potential solution. The algorithm is divided into rounds, where after each round, some chromosomes are removed, a process inspired by natural selection. Which chromosomes are removed is dictated by a heuristic, that indicates how close a chromosome is to being a solution. Surviving chromosomes are altered by performing crossover and mutation, to form a new population for the next round.

The other one is an iterative, heuristic algorithm with 2 parameters. Parameter t dictates, the maximum number of iterations in a row, which do not improve total score. Parameter i specifies the number of nodes with are removed and added on each iteration.

3 (Team) Orienteering Problem with Time Windows

3.1 Definition

The Orienteering Problem with Time Windows expands the normal OP and now considers time window constraints that arise when the service at a particular node has to start within a predefined time window. Each node can only start in the assigned time window $[O_i, C_i]$ and arriving outside the given time window leads to inaccessibility or waiting time [4]. The TOPTW is proven to be NP-hard so its unlikely that it can be solved to optimality withing polynomial time [2].

3.2 Solution approaches

3.2.1 Iterated local search

[7] proposes a algorithm to obtain high quality results for the TOPTW in very limited computation time by speeding up the evaluation of possible improvements and better exploring the whole solution space. This is done by a fast iterated local search heuristic which combines a insertion step and a shaking step. The insertion step tries to add new nodes to a tour and verifies that the time windows are still satisfied after insertion. The nodes are selected based on their $Ratio_i = (S_i)^2/Shift_i$. The Shift is the total time consumption to insert an extra visit between two other visits. In the shaking step, one or more nodes will be removed and the following nodes shifted towards the start. This is done to escape local optima. After all parameters are initialized, the heuristic follows a simple loop of the insertion step until there are no new improvements to the solution for a fixed time. The current solution is recorded, and the shake step applied. Even with limited computation time, the algorithm provides high quality results.

[7]

3.2.2 SAILS

The SAILS algorithm is proposed by [3] and is a hybridization of Simulated Annealing and Iterated Local Search. Iterated Local Search is a very simple but effective algorithm, but since it accepts only improving solutions, it can not escape local optimums. To solve this Problem, Simulated Annealing was included to accept worse solutions with a changing probability. Instead of starting with a random solution, the algorithm uses a greedy construction heuristic. This Solution is then further improved with the SAILS algorithm. Computational results showed that SAILS is competitive to other state-of-the-art algorithms

3.2.3 LP-based Granular Variable Neighborhood Search

A Variable Neighborhood search scheme hybridized with the idea of exploring, most of the time, granular instead of complete neighborhoods is developed in [4]. The Aim was to increase the efficiency without losing the effectiveness of the algorithm. The approach is based on an idea that longer edges usually have a lower likelihood to belong to optimal solutions in routing problems. So by removing all edge which are longer than a given threshold, the algorithm avoids visiting several non-promising solutions. Variable Neighborhood Search is a simple approach based on the idea of changing a neighborhood every time the algorithm reaches a local optimum. A neighborhood consists of all the solutions which can be obtained by randomly removing a fixed amount of consecutive nodes from the current one. The granular variant of Variable Neighborhood Search tries to reduce the size of analyzes neighborhoods by excluding non promising arcs. Computational results showed that Variable Neighborhood Search is already an effective approach, and the addition of granularity can improve efficiency while maintaining effectiveness. On average, the algorithm performs quite well compared to state-of-the-art algorithms

4 Multi-Constraint Orienteering Problem with Multiple Time Windows

The Multi-Constraint Orienteering Problem with Multiple Time Windows (MC-OP-MTW) extends the OPTW by proposing two new components to add to the existing problem.

First, in the original OP, trips are only constrained using one budget, which is usually described to be the time that can be allocated to the trip. This budget is used up by travelling between the locations. In the multi-constraint extension of the problem, trips can have multiple budgets at the same time. For example, a trip could be limited by a temporal and monetary budget. This allows us to better reflect, what trips are actually possible in the real world.

Second, it enables a location to have multiple time windows, in which it can be visited. This allows to model e.g. a location having a break at noon where it is closed.

The paper proposes an approximating solution for the problem which uses a Simulated annealing heuristic to find a local optimum for the trip. Simulated annealing is a specialized type of local search algorithm which, unlike greedy local search algorithms, can escape local maximums to find better solutions. The probability of allowing a score to get worse can be controlled with a parameter, which allows the user to make a dynamic trade-off between run-time and result quality.

5 Tourist Trip Design Problem

5.1 Definition, use cases and goal

The *Tourist Trip Design Problem* (TTDP) is an interesting application for the OP that has been investigated in many current research articles and is often used to model several variants of the OP, e.g. the (T)OPTW or the MCTOPMTW. It describes the scenario in which a tourist wants to visit as many interesting sights or tourist attractions (also called points of interest or POIs) as possible without violating their time and budget constraints. In practice, the TTDP comes into play, for example, when designing *Personalized electronic tourist guides* (PET) [1].

The goal of the TTDP is – similar to the OP – to maximize the overall collected profit without violating the time constraint (and other constraints in case of the TTDP, as in multi-constraint variants of the OP). A high quality solution for the TTDP should take recommendations and objective ratings of POIs as well as (subjective) individual user preferences into account and should schedule all POI visits in a feasible or even nearly optimal route [1].

Gavalas et al. [1] conducted a survey on existing research on the TTDP with special focus on more realistic variants of the OP that capture the complex aspects of the TTDP.

5.2 Models for the TTDP

The general TTDP can be divided into two categories:

- single tour variants
- multiple tour variants

The single tour variants can be modeled via single-criterion variants of the *traveling salesman problem with profits* (TSPP), e.g. the OP. However, the plain OP describes a rather abstract scenario; thus, in practice the TDPP is often modeled by more realisitic variants of the OP. One such variant of the OP that adds time windows to POIs is the *OP with time windows* (OPTW). There are also other variants of the TSPP that can be used to model the TTDP; though, the OP is best suited because it rather focuses on maximizing the collected profit than on minimizing the travel cost. Time windows are especially useful to model opening times of POIs. Multiple tour variants of the TTDP can be described in terms of variants of the *vehicle routing problem with profits* (VRPP); in particular, the *team OP* (TOP) and its extensions. One such extension is the *multi-constraint team orienteering problem with time windows* (MCTOPTW), which – additionally to the time windows – considers multiple constraints when generating a route along POIs. This is useful in the TDPP because tourist may, in addition to time constraints, also have budget constraints or even other constraints. As for the single TTDP variants, there are also other variants of the VRPP that can be used to model the TTDP (the TOP is still the best suited one) [1].

Other than dividing the TTDP in single and multiple tour variants, it is also possible to divide the problem in the categories *undirected* and *directed* (if edge direction is considered in the underlying graph) [1].

Gavalas et al. [1] investigated several algorithms that try to solve the OP exactly or at least try to approximate it. In reality, however, it is often more useful to use heuristics in order to find sufficient solutions for the OP.

Other variants of the OP that are useful to model several aspects of the TTDP are, e.g., the *generalized OP* (GOP) (see 2), the *multi-objective OP* (MOOP) (where POIs can be assigned to different categories and provide different benefits based on the categories) or the *stochastic OP* (SOP) (where visit times of POIs are not known before the visit is completed).

In general, many current research articles on the TTDP often formulate the TTDP in terms of the OP or based on its extensions.

5.3 Solvers for the TTDP

Lim et al. [5] model the TTDP in terms of the OP with several realistic extensions and propose an algorithm *PersTour* for recommending personalized tours based on user interests derived from real travel histories, which are obtained from geo-tagged photographs. The advantage of *PersTour* in comparison to previous solutions is that it uses up-to-date travel data, prioritizes more recent travel data over past data, recommends tour sequences with personalized visit durations for POIs, updates recommendations based on the actual route the user has chosen so far and relies on time-based user interest rather than frequency-based user interest.

In particular, *PersTour* outperforms established algorithms like collaborative filtering recommender systems (that recommend POIs based on user similarities) and greedy-based approaches by tour popularity, user interest, precision and F_1-score [5].

The framework proposed by Lim et. al [5] obtains geo-tagged photograph with coordinates and time taken from Wikipedia or Flickr. Further, the framework divides into the following three steps:

1. Determination of POI visits (extract POI data from photographs)

2. Construction of travel history/sequences based on data from *Step 1*

3. Compute tour recommendation with *PersTour* algorithm

Another solution for the TTDP that has been implemented in a practical web application has been proposed by Woerndl et al. [9]. A user enters start and end points as well as preferences over a set of POI categories into a web application and receives a recommendation for a walking route along a sequence of POIs. The underlying algorithm to generate walking routes is a modified version of the Dijkstra algorithm; instead of computing the shortest path between start and end point, the algorithm computes the most interesting path along POIs.

The general framework divides into the following steps [9]:

1. Retrieval and scoring of POI items

2. Combination and grouping of items to composite trip

The retrieval of POI items is done via the Foursquare API. The web application retrieves POI data in a circular area around the mid between the start and end points entered by the user. After that, the retrieved items are sorted into six different POI categories. Finally, the score is applied [9].

The combination of POIs to form a composite trip is done via the modified Dijkstra algorithm. Woerndl et al. [9] have implemented a constraint-free and constraint based version of the algorithm. The constraint-free variant tries to maximize the fraction *entertainment/distance* instead of computing the shortest path between start and end point. For further improvement, the algorithm takes into account user preferences and underrepresented categories in the set of discovered places. The constraint-based algorithm, on the other hand, ensures that users time and budget constraints are not violated and, hence, only maximizes *entertainment* while considering user preferences and underrepresented categories as well.

In order to validate their approach other than only for optimization towards shortest paths among POI sequences or quickest traversal of tourist locations, Woerndl et al. [9] tested the user acceptance of their web application.

The authors also outline several challenges for future work; for example, balancing out the number of places per category, allowing users to customize categories and provide preferences for that customized categories or applying machine-learning techniques to adapt heuristics based on submitted feedback [9].

References

[1] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20:291–328, 2014.

[2] Bruce L Golden, Larry Levy, and Rakesh Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318, 1987.

[3] Aldy Gunawan, Hoong Chuin Lau, and Kun Lu. Sails: hybrid algorithm for the team orienteering problem with time windows. 2015.

[4] Nacima Labadie, Renata Mansini, Jan Melechovský, and Roberto Wolfler Calvo. The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research*, 220(1):15–27, 2012.

[5] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekara. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems*, 54:375–406, 2018.

[6] John Silberholz and Bruce Golden. The effective application of a new approach to the generalized orienteering problem. *Journal of Heuristics*, 16(3):393–415, June 2010.

[7] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Computers Operations Research*, 36(12):3281–3290, 2009. New developments on hub location.

[8] Xia Wang, Bruce L. Golden, and Edward A. Wasil. Using a Genetic Algorithm to Solve the Generalized Orienteering Problem. In Bruce Golden, S. Raghavan, and Edward Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, Operations Research/Computer Science Interfaces, pages 263–274. Springer US, Boston, MA, 2008.

[9] Wolfgang Wörndl, Alexander Hefele, and Daniel Herzog. Recommending a sequence of interesting places for tourist trips. *Information Technology and Tourism*, 17:31–54, 2017.