technische universität
dortmund

Masterthesis

Customizable Roundtrips with Tour4Me

Meta-heuristic Approaches for Personalized Running and Cycling Routes

Lisa Salewsky
May 2024

Supervisors:
Prof. Dr. Kevin Buchin
Mart Hagedoorn, M. Sc.

Technische Universität Dortmund
Fakultät für Informatik
Algorithm Engineering (LS-11)
http://ls11-www.cs.tu-dortmund.de

Contents

1 Introduction **1**

1.1 Goal and Methodology . . . 2

1.2 Structure . . . 3

2 Fundamentals and Background **5**

2.1 Shortest Path algorithms . . . 5

2.2 Meta-heuristics . . . 5

2.2.1 Ant Colony . . . 5

2.2.2 Genetic Algorithms . . . 5

2.2.3 Simulated Annealing . . . 5

3 Related Work **7**

3.1 Tour4Me . . . 7

3.2 Roundtrip paths . . . 9

3.2.1 RouteLoops & RouteYou . . . 9

3.2.2 Computing Running Routes . . . 10

3.2.3 Computing Cycling Routes . . . 11

4 Implemented Changes **13**

4.1 Application . . . 13

4.1.1 New Architecture . . . 13

4.1.2 Database . . . 13

4.1.3 Interface and Frontend changes . . . 13

4.2 Algorithmic changes . . . 13

4.2.1 Ant Colony . . . 13

4.2.2 Genetic Algorithms . . . 13

4.2.3 Simulated Annealing . . . 13

4.3 Parameter changes . . . 13

5 Evaluation **15**

6 Conclusion **17**
6.1 Results . . . 17
6.2 Future Work . . . 17

A Source Code **19**

Bibliography **i**

List of Figures **iii**

List of Algorithms **v**

Affidavit **vii**

Chapter 1

Introduction

Algorithms for optimal tours are an important and much studied part of computer science. It is a topic that directly influences the lives of many people. Better routing algorithms can help reduce travel times by car, bicycle or even on foot. There is also considerable work on optimizing public transportation [2, 11] and managing traffic jams [9, 10]. Examples are Dijkstra (uni- and bidirectional) [17, 25, 30], A* search (also uni- and bidirectional) [17, 25, 30], greedy algorithms [17, 30], branch-and-bound algorithms [16], the Bellman-Ford-Moore algorithm [8] and many more [12, 25].

All of these have in common that they always look for the shortest or quickest path between two different points. However, when planning a tour, the goal might not be to simply get to a location as quick as possible. In particular, in many cases people plan round-trips. Especially for running and cycling - for training towards a specific goal or even as a pastime hobby - it is often desired to have roundtrips of a certain length. Additionally, people typically enjoy running or cycling on more appealing paths in nature rather than between high buildings and on softer ground rather than on asphalt. So, a lot more information have to be taken into account when trying to find good running or cycling roundtrips. Which means, these algorithms become useless for these scenarios. The shortest path from the starting point back to it will always be to never leave. So, a different approach is needed for these kinds of routes [13].

Considering the benefits of running and cycling (for overall health [21, 24, 27], the cardiovascular system[19], as a measure against many different diseases[21] as well as for social[18, 20, 29] and psychological benefits[3, 7, 26, 29] and also further benefits of touristic cycling for cities [4]), the problem at hand becomes all the more important. Not only are there many joggers and cyclists, who would profit from a tool that returns a roundtrip for their preferred personal optimal route, but having such a tool at hand could help convince more people of starting one or both of the two activities. This could result in an overall larger population doing some exercise and profiting from the previously mentioned benefits of physical activity outdoors. Not only does such a web app lower the effort it takes to start running or cycling (as route planning is coupled with effort), it also helps to show people better or more appealing routes and encourage participation in outdoor activities.

Additionally, as already stated in examples for benefits of running and cycling, such an app can prove useful for tourism purposes as well. People typically enjoy running or cycling along enticing, exiting routes, which are hard to find - especially in unfamiliar areas. For any kind of holiday trip, planning new roundtrips for either exercise purposes or event for several-day roundtrips, this app can be very useful.

1.1 Goal and Methodology

The goal of this thesis is to create a usable application for computing running or cycling roundtrips of (almost) arbitrary length. Useful in this case means an app that can be used in real time, that produces results of the desired length and prioritizes paths according to the users' input. To achieve this, the thesis will be built on the already existing prototype Tour4Me and eventually add meta-heuristic approaches that have been deemed the most fitting for its purpose.

First, an interface for testing the new approaches has to be built. This also needs an overlay for adding in user options like the length of the desired roundtrip, as well as other preference inputs. Based on this interface, different algorithms can be added and compared with each other to find the ones that will produce optimal results. These optimal results can have very different definitions of optimal. An ideal algorithm would be fast, always generate a route and use all the users' preference inputs. However, it is not possible to achieve all these goals with just one algorithm. Therefore, different approaches will be implemented and analyzed according to how well they fulfill the previously mentioned criteria.

Some of the possible approaches include different implementations of genetic algorithms [14], of ant colony or anthill algorithms [1, 14, 28] as well as possible hybrid versions. These hybrids can either be hybrids of one of the meta-heuristics with - for example - local search algorithms [14, 28] or hybrids of these two joined together. Furthermore, if there is enough time left, it is also possible to include the new algorithms into already implemented ones to improve those.

When the best algorithms for this application have been determined, they will be integrated into the already existing Tour4Me application. The aim is for the app to calculate a high-quality tour for any typical roundtrip requests for running and cycling.

In addition to finding suitable algorithms that allow for fast and reliable computation of all typical roundtrips, working on the interface and data used also improves the usefulness of the app. It can be equally important to improve the interface, add more options like elevation data, include more information (for example previously used routes) etc. There are several opportunities and options to improve the app not only by changing the used algorithms but also adding and upgrading the GUI and user selection options. This is an alternative approach towards the goal of making Tour4Me more usable. It is another option to put more work into improving the app aside from adding more or faster algorithms.

1.2 Structure

Chapter 2

Fundamentals and Background

As stated in the introduction, most routing algorithms focus on shortest paths between two or more points. Many of those have been reviewed in several different surveys [17, 25, 30]. Additionally, there have been many more heuristic approaches, like local search variants [5, 15, 23] or different neighborhood based ideas [5, 15, 23]. Much research has been done and is still ongoing for these kinds of problems, which is only natural since many routing problems (for example the traveling salesman problem (TSP) [14] or the vehicle routing problem [5, 15]) are NP-hard [22].

2.1 Shortest Path algorithms

2.2 Meta-heuristics

2.2.1 Ant Colony

2.2.2 Genetic Algorithms

2.2.3 Simulated Annealing

Chapter 3

Related Work

Much research has been done for shortest path algorithms and their optimization, however, for the - more complicated [13] - problem of finding a round trip with additional conditions, not much work has been done yet. While there are a few tools that can be used to calculate round trips, most of them only focus on cycling or create a very limited set of trips that do not satisfy the needs of most people, or both. Some examples for these tools are RouteLoops [1] and RouteYou [2] which both do not allow for much customization of preferences.

Adding new options for user inputs that enable a higher degree of customization can vastly improve the usability of a tool. The usefulness is not only determined by the implemented algorithms, but also by the interface, the data used, and the selection options presented to the user.

As both RouteLoops and RouteYou are commercial programs, it was not possible to obtain the necessary details about any used algorithms, heuristics, meta-heuristics or even the language they used for programming these solutions. All gathered information are collected from exploring the functionality of the two tools by hand and reading both the general information and the FAQ pages provided by the websites.

3.1 Tour4Me

The tool which this thesis will be based off, Tour4Me[3] [6], incorporates many of these points in its web interface. It is possible to choose the preferred ground type as well as make selections about preferred route types. Furthermore, the user can also mark certain types as unpreferrable (rather than just keeping them neutral or preferring them). This allows for much more customization. What the tool does not incorporate yet is the option to make selections about the preferred elevation or route complexity. However, the route can be optimized for a circular route when using the covered area of the tour.

[1] https://www.routeloops.com/
[2] https://www.routeyou.com
[3] http://tour4me.cs.tu-dortmund.de/

It implements a solution for the "touring problem", which is used to describe the task of finding appealing and ideally interesting roundtrips. To achieve an optimal solution, two factors are taken into consideration. First is the total profit, that can be collected within the given length restriction for the tour. Second is an additional quality function that assures for a relatively round tour by maximizing the area that is surrounded by the created roundtrip. Tour4Me presents a selection of four different algorithms to calculate the tour as well as some additional customization options. The offered choices include a Greedy Selection approach, Integer Linear Programming, MinCost with Waypoints, a shortest paths variant, and Iterative Local Search [6].

The Greedy Selection is the simplest algorithm which only ensures that the chosen route is a roundtrip. It builds it's path by iterating over the valid edges and picking the most profitable of these until the cycle is finished or no candidate is left. A valid edge is determined by checking whether the start- and endpoint s can still be reached if that edge is picked next [6].

For Integer Linear Programming, the touring problem must be stated in an appropriate form. To do so, a single instance can be encoded as $\mathcal{I}(G, w, \pi, B, v_0)$, containing the Graph G, edge costs w, the profit function π, the budget (length restrictions) B and the starting (and end-) point v_0. Given this encoding, cycles $P = (v_0, ..., v_i, ..., v_0)$, which are always at most of length L, can be built. For the current definition, a few additional variables an be introduced to encode whether or not an edge is part of a solution (and how many times it occurs), whether or not an edge is the k-th edge of the solution and whether or not a vertex is the k-th vertex of a solution. Using these, constraints can be built to describe the desired behavior of the algorithm [6].

The MinCost algorithm needs the waypoints because it is typically meant to solve shortest path problems. Thus it would always choose not leaving the starting position without the added points. Even though this algorithm is not originally meant to solve roundtrip problems, it takes into account the cost and profits of edges to create a solution tour, which makes it more suited to the task than simple greedy search. To create an optimized tour, the inefficiency of paths has to be measured. This is done by calculating the quotient of the edge costs and the profit the edge yields. Using this inefficiency, a ring of candidate points R_s surrounding the start-point s can be calculated. All points that are part of this ring have a shortest path distance of at most π. From these, new rings R_r with the same requirements can be calculated. The solution path is then obtained through intersecting the sets of all circles and selecting all those that intersect with R_s. To ensure the highest profit tour is returned, all possible combinations are calculated and the optimum is returned[6].

Building from this solution, the Iterative Local Search can be applied to improve the found tours. From the returned roundtrip, the algorithm removes partial paths P from the current best solution S and tries to iteratively add new parts that improve the solution profit while always staying within the given budget $(B - w(\frac{S}{P}))$. Since searching for viable edges is performed using a depth first approach, bounding the maximum depth of this step can drastically speed up the algorithm. To keep track of the added length and profit, two variables (l and p respectively) are introduced. These start with an initial value of one and are raised by a single increment for each iteration. p is reset when the starting point is reached by the removal step. l is reset when the maximum length for the solution is reached. The best solution is improved constantly until the user selected time limit is reached [6].

#TODO add more citations –> see Tour4Me paper

3.2 Roundtrip paths

As already stated above, existing tools leave out certain data like elevation or path types. This impacts the quality of the created routes for users or even user groups. For example, people who prefer running with little to no elevation can end up with a route that takes them uphill through a park for half of the route. Which still may be a good choice for other users - joggers who prefer more challenging routes or people who want to hike and enjoy ascending. However, others could prefer running through the city over a park when the elevation matches their preferences better in the city. For these users, the created route would be highly unfavorable, even though it matches other constraints for what is considered a nice roundtrip. Therefore, it can be crucial to the usefulness of an app to give the user as many options to customize as possible.

3.2.1 RouteLoops & RouteYou

RouteLoops has two text fields for entering the starting point and the length of the trip. Aside from that, no real customization is possible. It does have a few features to show more information about the route like showing distance markers or elevation, however, these can not be used as inputs to get a route with - for example - as little elevation as possible. Apparently it can also show route difficulty for the United States, however even when creating a route in the United States, no result was shown. RouteLoops also does not actually create loops but rather picks a route that has high value (for example with a river in a park) and lets the user run along that path, turn around at the end and run back the same way.

To crate a roundtrip, some "waypoints" are created. These can be removed or more can be added in when editing the tour. Between the waypoints, it seems like a shortest path is tried

RouteYou offers several different options that will return varying results, however, picking the same option again will also give different results every time. Here, the roundtrips are more round than with RouteLoops, but again, elevation or difficulty are not taken into account. Also, while both do offer the possibility to edit the returned roundtrip, this editing changes the length of the route arbitrarily. Furthermore, it is not possible to specify directly what type of underground or surroundings etc. are preferred.

3.2.2 Computing Running Routes

The problem of calculating good running roundtrips is not new. In addition to the commercial applications, there also are research papers on this subject. One of these papers is "Efficient Computation of Jogging Routes" [13], presents two ideas to handle the new routing problem which the authors labeled "Jogging Problem". It is split up into two variants: One being the simple version, that only aims to build a cycle that contains the starting point s and has the desired length. The other is a more complex version, that allows for some flexibility regarding the length of the final tour during optimization. Hence, it is named "Relaxed Jogging Problem". This relaxation allows to take more factors into account to also optimize for the resulting shape, the area surrounding the tour and/or the simplicity of the path [13].

The second problem is chosen as the one to optimize, since it enables the addition of other conditions than just the length of a tour. For this, two different ideas are proposed. The first approach - "Greedy Faces" - is based on extending previous cycles. It starts with a cycle containing the starting point s that can be selected by the user. This roundtrip then can be extended to gradually approach the user specified length. The second algorithm was named "Partial Shortest Paths" and uses via-vertices. These are a number of new points that can be connected with shortest paths. When the via-vertices are connected with each other and the start, they form a roundtrip [13].

For both algorithms, the authors measure the badness of paths, the number of edges that are shared as well as the number of turns. The badness is used to take the additional constraints into account. To reduce the possibility of having a roundtrip which turns at the end and uses all paths twice, the shared edges have to be minimized. The number of turns corresponds to the complexity of the tour and is measured by a percentage of a full u-turn [13].

#TODO how often do I need to reference the paper?

Greedy Faces Greedy Faces is built from an already existing circle by extending it. For this, blocks outside the tour that are adjacent to the current path are used. The previous circle encloses the chosen block and thus extends the previous route. New blocks are picked until the desired length is reached. To ensure only blocks that correspond to faces are picked, a preprocessing phase is introduced that identifies faces of the graph. During this step, first, dead-ends are removed, so the resulting graph will be two-connected. Faces then are defined by the edges that surround them. While identifying all faces, a dual graph $G^\star = (V^\star, E^\star)$ for $G = (V, E)$ is built as well.

The Greedy Faces algorithm then works on the dual graph $G^\star$, selects a face f from $V^\star$ which has a surrounding path that contains the starting point s. Then, a Breadth First Search Tree T is built, starting at f, until the desired length (a relaxed version $(1+\varepsilon)L$) is exceeded. The resulting tour will be a simple path iff all vertices in V without the ones in T are connected and contain s. The final jogging path can be extracted by taking the cut edges between the tree T and the remaining vertices. This always forms a circle and thus builds a roundtrip.

For building a path which optimizes all constraints, the three introduced measures for badness, number of shared edges and the number of turns are used. The badness function is incorporated into a different force function which can assign positive and negative badness values to edges. Furthermore, the force function uses the cost of the face and a vector $\vec{d} = \vec{p} - \vec{C}(f)$ which is built from the geometric center $\vec{C}(f)$ of a face to any point $\vec{p}$

After the tour has been created, it will be smoothed to reduce the complexity.

Partial Shortest Paths

Computational Complexity Aside from introducing two methods to calculate roundtrip tours for running, this paper also presents a proof for the computational complexity of their Simple and Relaxed Jogging Problems. The authors show NP hardness by reduction of Hamiltonian Cycle to the optimization problem corresponding to their original problems.

TODO add the actual proof?

TODO other jogging route paper

3.2.3 Computing Cycling Routes

TODO cycling paper Tour4Me

TODO other cycling paper

Chapter 4

Implemented Changes

4.1 Application

4.1.1 New Architecture

4.1.2 Database

4.1.3 Interface and Frontend changes

4.2 Algorithmic changes

4.2.1 Ant Colony

4.2.2 Genetic Algorithms

4.2.3 Simulated Annealing

4.3 Parameter changes

Chapter 5

Evaluation

Chapter 6

Conclusion

6.1 Results

6.2 Future Work

Appendix A

Source Code

Bibliography

[1] Babaoglu, O., H. Meling and A. Montresor: *Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems.* In *Proceedings 22nd International Conference on Distributed Computing Systems*, pages 15–22, Vienna, Austria, July 2002. IEEE.

[2] Bast, Hannah, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner and Renato F. Werneck: *Route Planning in Transportation Networks.* In Kliemann, Lasse and Peter Sanders (editors): *Algorithm Engineering: Selected Results and Surveys*, Lecture Notes in Computer Science, pages 19–80. Springer International Publishing, Cham, 2016.

[3] Biddle, Stuart J. H.: *Psychological benefits of exercise and physical activity.* Revista de psicología del deporte, 2(2):0099–107, 1993.

[4] Blondiau, Thomas, Bruno van Zeebroeck and Holger Haubold: *Economic Benefits of Increased Cycling.* Transportation Research Procedia, 14:2306–2313, January 2016.

[5] Bräysy, Olli and Michel Gendreau: *Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms.* Transportation Science, 39(1):104–118, February 2005.

[6] Buchin, Kevin, Mart Hagedoorn and Guangping Li: *Tour4Me: A Framework for Customized Tour Planning Algorithms.* In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–4, Seattle Washington, November 2022. ACM.

[7] Cekin, Resul: *Psychological Benefits of Regular Physical Activity: Evidence from Emerging Adults.* Universal Journal of Educational Research, 3(10):710–717, October 2015.

[8] Cherkassky, Boris V., Andrew V. Goldberg and Tomasz Radzik: *Shortest Paths Algorithms: Theory and Experimental Evaluation.* Mathematical Programming, 73(2):129–174, May 1996.

[9] Delling, Daniel: *Time-Dependent SHARC-Routing.* Algorithmica, 60(1):60–94, May 2011.

[10] Delling, Daniel, Andrew V. Goldberg, Thomas Pajor and Renato F. Werneck: *Customizable Route Planning in Road Networks.* Transportation Science, 51(2):566–591, May 2017.

[11] Delling, Daniel, Thomas Pajor and Renato F. Werneck: *Round-Based Public Transit Routing.* Transportation Science, 49(3):591–604, August 2015.

[12] Delling, Daniel, Peter Sanders, Dominik Schultes and Dorothea Wagner: *Engineering Route Planning Algorithms.* In Lerner, Jürgen, Dorothea Wagner and Katharina A. Zweig (editors): *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*, Lecture Notes in Computer Science, pages 117–139. Springer, Berlin, Heidelberg, 2009.

[13] Gemsa, Andreas, Thomas Pajor, Dorothea Wagner and Tobias Zündorf: *Efficient Computation of Jogging Routes.* In Bonifaci, Vincenzo, Camil Demetrescu and Alberto Marchetti-Spaccamela (editors): *Experimental Algorithms*, Lecture Notes in Computer Science, pages 272–283, Berlin, Heidelberg, 2013. Springer.

[14] Gendreau, Michel and Jean-Yves Potvin (editors): *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science.* Springer US, Boston, MA, 2010.

[15] Irnich, Stefan, Birger Funke and Tore Grünert: *Sequential Search and Its Application to Vehicle-Routing Problems.* Computers & Operations Research, 33(8):2405–2429, August 2006.

[16] Lawler, E. L. and D. E. Wood: *Branch-and-Bound Methods: A Survey.* Operations Research, 14(4):699–719, August 1966.

[17] Madkour, Amgad, Walid G. Aref, Faizan Ur Rehman, Mohamed Abdur Rahman and Saleh Basalamah: *A Survey of Shortest-Path Algorithms*, May 2017.

[18] Mueller, Florian 'Floyd', Shannon O'Brien and Alex Thorogood: *Jogging over a Distance: Supporting a "Jogging Together" Experience Although Being Apart.* In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA

'07, pages 2579–2584, New York, NY, USA, April 2007. Association for Computing Machinery.

[19] Nystoriak, Matthew A. and Aruni Bhatnagar: *Cardiovascular Effects and Benefits of Exercise.* Frontiers in Cardiovascular Medicine, 5:135, 2018.

[20] O'Brien, Shannon and Florian "Floyd" Mueller: *Jogging the Distance.* In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 523–526, San Jose California USA, April 2007. ACM.

[21] Oja, P., S. Titze, A. Bauman, B. de Geus, P. Krenn, B. Reger-Nash and T. Kohlberger: *Health Benefits of Cycling: A Systematic Review.* Scandinavian Journal of Medicine & Science in Sports, 21(4):496–509, 2011.

[22] Reinelt, Gerhard: *The Traveling Salesman: Computational Solutions for TSP Applications.* Springer, Berlin, Heidelberg, August 2003.

[23] Ropke, Stefan: *Heuristic and Exact Algorithms for Vehicle Routing Problems.* PhD thesis, Technical University of Denmark, January 2005.

[24] Ruegsegger, Gregory N. and Frank W. Booth: *Health Benefits of Exercise.* Cold Spring Harbor Perspectives in Medicine, 8(7):a029694, January 2018.

[25] Sommer, Christian: *Shortest-Path Queries in Static Networks.* ACM Computing Surveys, 46(4):1–31, April 2014.

[26] Szabo, Attila and Júlia Ábrahám: *The Psychological Benefits of Recreational Running: A Field Study.* Psychology, Health & Medicine, 18(3):251–261, May 2013.

[27] Vina, J, F Sanchis-Gomar, V Martinez-Bello and Mc Gomez-Cabrera: *Exercise Acts as a Drug; the Pharmacological Benefits of Exercise.* British Journal of Pharmacology, 167(1):1–12, 2012.

[28] Wang, Xueyang, Chonghua Liu, Yupeng Wang and Chengkai Huang: *Application of Ant Colony Optimized Routing Algorithm Based on Evolving Graph Model in VANETs.* In *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 265–270, Beijing, China, September 2014. Institute of Electrical and Electronics Engineers (IEEE).

[29] Wankel, Leonard M. and Bonnie G. Berger: *The Psychological and Social Benefits of Sport and Physical Activity.* Journal of Leisure Research, 22(2):167–182, April 1990.

[30] Wayahdi, Muhammad Rhifky, Subhan Hafiz Nanda Ginting and Dinur Syahputra: *Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path.* International Journal of Advances in Data and Information Systems, 2(1):45–52, February 2021.

List of Figures

Algorithmenverzeichnis

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kentlich gemacht habe.

Dortmund, den December 17, 2023

Muster Mustermann