Speedups for Multi-Criteria Urban Bicycle Routing

Jan Hrncir, Pavol Zilecky, Qing Song, and Michal Jakob

Agent Technology Center, Department of Computer Science
Czech Technical University in Prague, Karlovo namesti 13, Czech Republic
{hrncir,zilecky,song,jakob}@agents.fel.cvut.cz

Abstract

Increasing the adoption of cycling is crucial for achieving more sustainable urban mobility. Navigating larger cities on a bike is, however, often challenging due to cities' fragmented cycling infrastructure and/or complex terrain topology. Cyclists would thus benefit from intelligent route planning that would help them discover routes that best suit their transport needs and preferences. Because of the many factors cyclists consider in deciding their routes, employing multi-criteria route search is vital for properly accounting for cyclists' route-choice criteria. Direct application of optimal multi-criteria route search algorithms is, however, not feasible due to their prohibitive computational complexity. In this paper, we therefore propose several heuristics for speeding up multi-criteria route search. We evaluate our method on a real-world cycleway network and show that speedups of up to four orders of magnitude over the standard multi-criteria label-setting algorithm are possible with a reasonable loss of solution quality. Our results make it possible to practically deploy bicycle route planners capable of producing high-quality route suggestions respecting multiple real-world route-choice criteria.

1998 ACM Subject Classification G.2.2 Graph Theory – Graph algorithms

Keywords and phrases bicycle routing, multi-criteria shortest path, heuristic speedups

Digital Object Identifier 10.4230/OASIcs.ATMOS.2015.16

1 Introduction

Utility cycling, i.e., using the bicycle as a mode of transport, is the original and the most common type of cycling in the world [13]. Cycling provides a convenient and affordable form of transport for most segments of the population. It has a range of health, environmental, economical, and societal benefits and is therefore promoted as a modern, sustainable mode of transport [10, 16].

In contrast to car drivers, cyclists consider a significantly broader range of factors while deciding on their routes. By employing questionnaires and GPS tracking, researchers have found that besides travel time and distance, cyclists are sensitive to slope, turn frequency, junction control, noise, pollution, scenery, and traffic volumes [3, 28]. Moreover, the relative importance of these factors varies among cyclists and can also be affected by weather conditions and the purpose of the trip [3].

Finding routes that properly take all the above factors into account is no easy task, particularly when cycling in complex urban environments. Consequently, cyclists would benefit from intelligent route planning software to help them discover routes that best suite their transport needs and preferences. Such route planners would be particularly useful for inexperienced cyclists with limited knowledge of their surroundings but they would also benefit experienced riders who want to fine-tune their routes [11], in effect making cycling a more attractive and accessible transport option.

The vast majority of existing approaches to bicycle routing, however, do not use multi-criteria search methods and they thus cannot properly account for cyclists' multiple route-

15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'15).
Editors: Giuseppe F. Italiano and Marie Schmidt; pp. 16–28
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

choice criteria. A recent exception is [24] where the authors applied optimal multi-criteria shortest path algorithms for multi-criteria bicycle routing. Unfortunately, the proposed algorithm is slow on realistic problem instances and cannot be used for interactive route planning.

In this paper, we overcome this limitation and present the first bicycle routing algorithm that properly considers multiple realistic cyclists' route-choice criteria yet is fast enough for interactive use. Our algorithm extends the well-known multi-criteria label-setting algorithm [19] with several speedup heuristics in order to generate, in a much shorter time, routes that closely approximate the full set of Pareto optimal routes. In contrast to the majority of existing work, our algorithm employs a formulation of the multi-criteria bicycle routing problem that incorporates realistic route choice factors based on recent studies of cyclists' behaviour [3, 28]. We thoroughly evaluate our algorithm in terms of the speed and quality of suggested routes on a diverse set of real-word urban areas.

2 Related Work

In contrast to car and public transport route planning, for which advanced algorithms and mature software implementations exist [1], bicycle route planning is a relatively underexplored topic. Furthermore, despite the highly multi-criterial nature of cyclists' route-choice preferences, almost all existing approaches to bicycle routing do not use multi-criteria search methods to properly account for such a multi-criteriality. This contrasts with other categories of routing problems for which the application of multi-criteria shortest path search techniques has been widely studied (multimodal routing [2, 7], train routing [20], and car routing [8]). Instead, existing bicycle routing approaches transform multi-criteria search to single-criterion search either by optimising each criteria separately [14, 26] or by using a weighted combination of all criteria [15, 27]. Unfortunately, the scalarisation of multi-criteria problems using a linear combination of criteria may miss many Pareto optimal routes [4, 6] and consequently reduce the quality and relevance of suggested routes. Scalarisation also requires the user to weight the importance of individual route criteria a priori, which is difficult for most users.

Avoiding scalarisation, [23] thus showed how to effectively search for a best compromise solution for a biobjective shortest path problem in the context of bicycle routing. Recently, [24] explored the use of optimal multi-criteria shortest path algorithms for multi-criteria bicycle routing; however, the proposed algorithm is too slow for interactive route planning.

As far as general multi-criteria shortest path algorithms are concerned, the first optimal, multi-criteria label-setting (MLS) algorithm [19] extended Dijkstra's algorithm by operating on labels that have multiple cost values. A minimum label from the priority queue is processed in every iteration. On the contrary, the multi-criteria label-correcting (MLC) algorithm [5, 9] processes the whole bag of nondominated labels associated with a current node at once. Recently, heuristic accelerations of the MLS and MLC algorithms have attracted considerable attention, aiming at finding a set of routes that is similar to the optimal Pareto solution. In [7], the authors developed several heuristics to weaken the domination rules during the search. In [22], the authors proposed a near admissible multi-criteria search algorithm to approximate the optimal set of Pareto routes in a state space graph by using the ϵ-dominance approach. An alternative approach is represented by multi-criteria extensions [18, 25] of the standard A* algorithm, the latter of which was recently shown [17] to achieve an order of magnitude speedup for bicriteria road routing.

3 Multi-Criteria Bicycle Routing Problem

We represent the cycleway network as a weighted directed connected *cycleway graph* $G = (V, E, \overrightarrow{c})$, where V is the set of nodes representing start and end points (i.e., cycleway junctions) of cycleway segments and $E \subseteq \{(u,v)|(u,v \in V) \wedge (u \neq v)\}$ is the set of edges representing cycleway segments. The cycleway graph is directed due to the fact that some cycleway segments in the map are one-way only. The cost of each edge is represented as a k-dimensional vector of criteria $\overrightarrow{c} = (c_1, c_2, \ldots, c_k)$. The non-negative cost value c_i of i-th criterion for the given edge $(u,v) \in E$ is computed by the cost function $c_i : E \to \mathbb{R}_0^+$. The *multi-criteria bicycle routing problem* is then defined as a triple $C = (G, o, d)$:

- $G = (V, E, \overrightarrow{c})$ is the *cycleway graph*
- $o \in V$ is the route origin
- $d \in V$ is the route destination

A route π, i.e., a finite path π with a length $|\pi| = n$ from the origin o to the destination d in the cycleway graph G has an additive cost value

$$\overrightarrow{c}(\pi) = \left(\sum_{j=1}^{|\pi|} c_1(u_j, v_j), \ldots, \sum_{j=1}^{|\pi|} c_k(u_j, v_j) \right)$$

The solution of the multi-criteria bicycle routing problem is a full Pareto set of routes $\Pi \subseteq \{\pi | \pi = ((u_1, v_1), \ldots, (u_n, v_n))\}$ non-dominated by any other solution (a solution π_p dominates another solution π_q iff $c_i(\pi_p) \leq c_i(\pi_q)$, for all $1 \leq i \leq k$, and $c_j(\pi_p) < c_j(\pi_q)$, for at least one j, $1 \leq j \leq k$).

Based on the studies of real-word cycle route choice behaviour [28, 3], we further consider a tri-criteria bicycle routing problem. The formulation of the problem is a compact version of the earlier formulation proposed in [24] and considers the following three route-choice criteria:

Travel time: The travel time criterion c_{time} reflects the duration in seconds of the cyclist's journey. Travel time is a sensitive factor in cyclists' route planning especially for commuting purposes. Our travel time calculation takes into account average cyclist's speed, uphills and downhills, quality of the road surface, and obstacles. To model the slowdown caused by obstacles such as stairs or crossings, we define the slowdown coefficient $r_{\text{slowdown}} : E \to \mathbb{R}_0^+$ which returns the slowdown in seconds on a given edge $(u,v) \in E$. For the case of uphill rides, we define the positive vertical ascend $a : E \to \mathbb{R}_0^+$ for a given edge $(u,v) \in E$ as

$$a((u,v)) := \begin{cases} h(v) - h(u) & \text{if } h(v) > h(u) \\ 0 & \text{otherwise} \end{cases}$$

where $h : V \to \mathbb{R}$ returns the elevation for each node $u \in V$. Analogously, for the case of downhill rides, we define the positive vertical descend $d : E \to \mathbb{R}_0^+$. We also define the positive descend grade $d' : E \to \mathbb{R}_0^+$ as $d'((u,v)) := \frac{d((u,v))}{l((u,v))}$ where $l((u,v))$ is the length of the edge $((u,v))$. To model the speed acceleration caused by vertical descend for a given edge $(u,v) \in E$, we define the downhill speed multiplier $s_d : E \to \mathbb{R}^+$ which depends on the positive descend grade d' and it is in the interval $[1, 2.5]$.

Considering the integrated effect of the edge length, the change in the elevation, and edge associated features, the travel time criterion is defined as

$$c_{\text{time}}((u,v)) = \frac{\text{distance}}{\text{speed}} + \text{slowdown} = \frac{l((u,v)) + a_l \cdot a((u,v))}{s \cdot s_d((u,v), s_{\text{dmax}}) \cdot r_{\text{time}}((u,v))} + r_{\text{slowdown}}((u,v))$$

where s is the average cruising speed of a cyclist and a_l is the penalty coefficient for uphill rides. The criteria coefficient $r_{\text{time}}((u,v))$ expresses the effect of a set of features $f((u,v))$ assigned to a given edge $(u,v) \in E$ with respect to travel time criterion.

Comfort: The comfort criterion c_{comfort} captures the preference towards comfortable routes with good-quality surfaces and dedicated cycleways or streets with low traffic. The cost function for the comfort is defined as

$$c_{\text{comfort}}((u,v)) = \max\{r_{\text{surface}}((u,v)), r_{\text{traffic}}((u,v))\} \cdot l((u,v))$$

where the surface coefficient $r_{\text{surface}}((u,v))$ penalises bad road surfaces, obstacles such as steps, and places where the cyclist needs to dismount his or her bicycle, with small values indicating cycling-friendly surfaces. The traffic coefficient $r_{\text{traffic}}((u,v))$ measures traffic volumes by considering the infrastructure for cyclists and the types of roads, where low-traffic cycleways are assigned a small coefficient value. The comfort is weighted by the edge length $l((u,v))$, i.e., 500 m of cobblestones is worse than 100 m of cobblestones.

Elevation gain: The elevation gain criterion c_{gain} captures the cyclists' preference towards flat routes with minimum uphill segments. The cost function is defined as

$$c_{\text{gain}}((u,v)) = \frac{\text{distance}}{\text{speed}} = \frac{a_l \cdot a((u,v))}{s}$$

where s is the average cruising speed of a cyclist, $a((u,v))$ is the positive vertical ascend of the edge (u,v), and a_l is the penalty coefficient for uphill rides.

4 Heuristic-Enabled Multi-Criteria Label-Setting Algorithm

Our newly proposed *heuristic-enabled multi-criteria label-setting* (HMLS) algorithm extends the standard multi-criteria label-setting (MLS) algorithm [19] with several points for inserting speedup heuristic logic. The algorithm uses the following data structures: for each node $u \in V$, $L(u) := (u, (l_1(u), l_2(u), \dots, l_k(u)), L^P(u))$ represents the *label* at a node u, which is composed of the node u, the cost vector $l(u)$ indicating the current cost values from the origin to the node u, and the predecessor label $L^P(u)$, which precedes $L(u)$ in an optimal route from an origin. A priority queue Q is defined to maintain all labels created during the search. Since each node may be scanned multiple times, we define the bag structure $Bag(u)$ for each node u to maintain the non-dominated labels at u.

The pseudocode of the heuristic-enabled MLS algorithm is given in Algorithm 1; the speedup specific logic of functions `terminationCondition`, `skipEdge`, and `checkDominance` is described in Section 5. The algorithm consists of the following steps:

Step 1 – Initialisation: For a k-criteria optimisation problem, the algorithm first initialises the priority queue Q and Bag for each $v \in V$. Then it initialises the label at the origin to $L(o) := (o, (l_1(o), l_2(o), \dots, l_k(o)), null)$, where $l_i(o) = 0$ for $i = 1, 2, \dots, k$. Finally, it inserts the initial label $L(o)$ into the queue Q and the $Bag(o)$.

Step 2 – Label expansion: The algorithm extracts a minimum label $current := (u, (l_1(u), l_2(u), \dots, l_k(u)), L^P(u))$ from the priority queue Q (in a lexicographic order of a cost vector). For each outgoing edge (u,v), the algorithm computes a new cost vector $(l_1(v), l_2(v), \dots, l_k(v))$ by adding the costs of the edge (u,v) to the current cost values $(l_1(u), l_2(u), \dots, l_k(u))$. Then,

Algorithm 1: Heuristic-enabled multi-criteria label-setting algorithm.

Input: cycleway graph $G = (V, E, \overrightarrow{c})$, origin node o, destination node d
Output: full Pareto set of labels

```
1  Q := empty priority queue
2  Bag(∀v ∈ V) := empty set
3  L(o) := (o, (0, 0, ..., 0), null)
4  Q.insert(L(o))
5  Bag(o).insert(L(o))
6  while Q is not empty do
7      current := Q.pop()
8      u := current.getNode()
9      (l_1(u), l_2(u), ..., l_k(u)) := current.getCost()
10     L^P(u) := current.getPredecessorLabel()
11     if terminationCondition(current) then
12         break
13     end
14     foreach edge (u, v) do
15         l_i(v) := l_i(u) + c_i(u, v) for i = 1, 2, ..., k
16         next := (v, (l_1(v), l_2(v), ..., l_k(v)), current)
17         if skipEdge(next) then
18             continue
19         end
20         if checkDominance(next) then
21             Bag(v).insert(next)
22             Q.insert(next)
23         end
24     end
25 end
26 return Bag(d)
```

it creates a new label *next* using the node v, the cost vector $(l_1(v), l_2(v), \ldots, l_k(v))$ and the predecessor label *current*.

Function `skipEdge` (cf. Algorithm 1, line 17) prevents looping the path by checking the predecessor label in the label data structure, i.e., if previous node $L^P(u).getNode()$ is equal to the node v then the edge (u, v) is skipped.

Function `checkDominance` (cf. Algorithm 1, lines 20–23), by default, controls dominance between the label *next* and all labels inside $Bag(v)$. If *next* is not dominated, the algorithm inserts it into $Bag(v)$ and Q. Also, if some label inside $Bag(v)$ is dominated by *next*, it is eliminated from the bag structure and not considered in future search.

Step 3 – Pruning condition: The algorithm exits if the queue Q becomes empty. Otherwise, it continues with *Step 2*.

After the algorithm has finished, the optimal Pareto set of routes Π^* is extracted. Let $|Bag(d)| = |\Pi| = m$. Then, from labels $L_1, \ldots, L_m$ in the destination Pareto set of labels

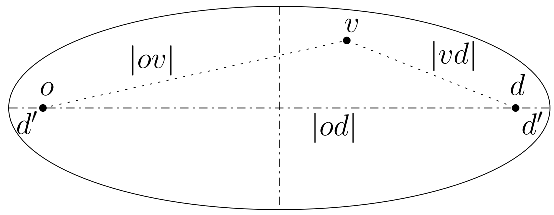

Figure 1 Geometry of the ellipse pruning condition.

$Bag(d)$, the routes $\pi_1, \ldots, \pi_m$ are extracted using the predecessor labels $L^P(\cdot)$. These routes comprise the set $\Pi^* = \{\pi_1, \pi_2, \ldots, \pi_m\}$ of optimal Pareto routes.

5 Speedups for the HMLS Algorithm

A significant drawback of the standard MLS algorithm is that it is very slow. The main parameter that affects the runtime of the algorithm is the size of the Pareto set. In general, the Pareto set can be exponentially large in the input graph size [21]. Furthermore, the MLS algorithm always explores the whole cycleway graph.

To accelerate the multi-criteria shortest path search, we introduce four speedup heuristics. Two of the heuristics are newly proposed by us: *ratio-based pruning* and *cost-based pruning*, while two are existing heuristics: *ellipse pruning* and *buckets*. Implementation-wise, the heuristics are incorporated into the heuristic-enabled MLS algorithm by defining the respective three heuristic-specific functions used in Algorithm 1.

Ellipse Pruning: The first speedup heuristic taken from [12] prevents the MLS algorithm from always searching the whole cycleway graph, even for a short origin-destination distance[1]. The heuristic permits visiting only the nodes that are within a predefined ellipse. The focal points of the ellipse correspond to the journey origin o and the destination d. Let $|od|$ be the direct origin-destination distance and d' the distance between origin and a peripheral point on the main axis of the ellipse. Then the length of the main axis $2a$ is equal to $|od| + 2d'$. During the search, in `skipEdge` function (cf. Algorithm 1, line 17), it is checked whether an edge (u, v) has its target node v inside the ellipse by checking the inequality $|ov| + |vd| \leq |od| + 2d'$, cf. Figure 1.

Ratio-Based Pruning: The ratio-based pruning terminates the search (long) before the priority queue gets empty (which means that the whole search space has been explored). A pruning ratio $\alpha \in \mathbb{R}^+$ is defined and the search is terminated when one of the criteria cost values, e.g., $l_1(u)$, in the *current* label exceeds α times the best so far value of the same criterion for a route that has already reached the destination (this is checked in the `terminationCondition` function, cf. Algorithm 1, line 11).

Cost-Based Pruning: The third heuristic we use does not expand the search to a label $L(v)$ which is very close in the cost space (criteria $c_1, \ldots, c_k$) to the existing non-dominated labels at the node v. The newly generated label $L(v)$ with a closer Euclidean distance than $\gamma \in \mathbb{R}^+$

[1] Note that in contrast with single-criterion Dijkstra's algorithm, the MLS algorithm does not stop when the destination node is first reached.

Table 1 Graph sizes for the experiments.

Graph	Nodes	Edges	Area
Prague A	9411	20420	Old Town, Vinohrady
Prague B	9665	20808	Strahov, Brevnov
Prague C	10652	24121	Liben, Vysocany

is discarded inside the `checkDominance` function (cf. Algorithm 1, line 20). Therefore, the search process is accelerated since fewer labels are inserted into the queue and the bag.

Buckets: The last heuristic defined in [7] discretizes the cost space using buckets for the criteria values. The heuristic is executed in the `checkDominance` function (cf. Algorithm 1, line 20). A function $bucketValue : \mathbb{R}_0^+ \rightarrow \mathbb{N}$ is used to assign a real cost value l_i an integer bucket value $bucketValue(l_i)$.

6 Evaluation

To evaluate our approach, we consider the real cycleway network of Prague. Prague is a challenging experiment location due to its complex geography and fragmented cycling infrastructure, which raises the importance of proper multi-criteria routing.

6.1 Experiment Setting

We evaluate our solution on cycleway graphs corresponding to three distinct areas of the city of Prague. We have chosen parts Prague A, Prague B, and Prague C to be different in terms of network density, nature of the cycling network and terrain topology so as to evaluate the performance of heuristics across a range of conditions. The sizes of the evaluation graphs are depicted in Table 1. The graphs are also shown in Figure 2 in the map of Prague. The specifics of each evaluation area are the following:

- Prague A: This graph covers a flat city centre area of the Old Town with many narrow cobblestone streets and Vinohrady with the grid layout of streets.
- Prague B: This graph covers a very hilly area of Strahov and Brevnov with many parks.
- Prague C: This graph covers residential areas of Liben and Vysocany further from the city centre. There are many good cyclepaths in this area.

All evaluation cycleway graphs are strongly connected. The size of evaluation graphs allows us to run the MLS algorithm without any speedups, which is crucial for comparing the quality of heuristic and optimal solutions.

For each graph evaluation area, a set of origin-destination pairs generated randomly with a uniform spatial distribution, was used in the evaluation. First, we generated 130 origin-destination pairs for each of graphs Prague A, B, and C. The minimum origin-destination distance is set to 500 m. The longest routes have approximately 4.5 km. From these 130 origin-destination pairs, we filtered out 15 pairs with the smallest size of the optimal Pareto set and 15 pairs with the largest size of the optimal Pareto set to receive a set of 100 origin-destination pairs. We executed the MLS algorithm and the HMLS with all 11 heuristic combinations using the same generated 100 origin-destination pairs for each graph Prague A, B, and C. Therefore, each heuristic combination is evaluated on 300 origin-destination pairs.

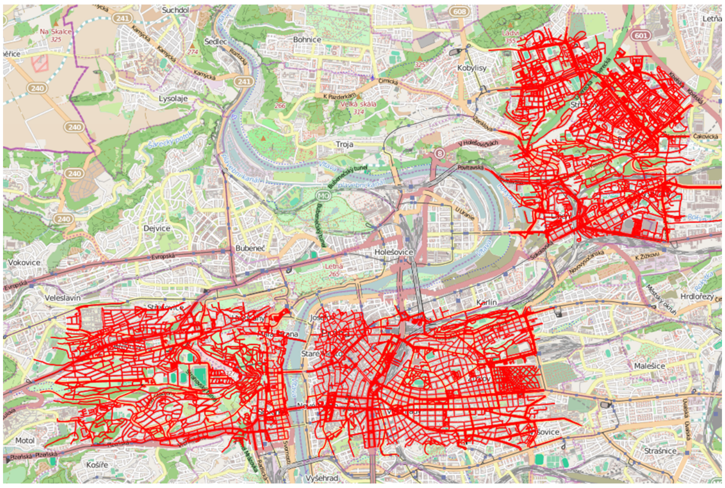

Figure 2 Evaluation graphs Prague B, Prague A, and Prague C (from left to right).

The parameters in the cost functions were set as follows. The average cruising speed is $s = 14$ km/h and the penalty coefficient for uphill is $a_l = 13$ (according to the route choice model developed in the user study [3]). Configuration parameters for the heuristics were set so as to maximize the ratio between the algorithm runtime and the quality of the solution (see the next section), as measured on the three graphs Prague A, B, and C. Specifically, the following values were used: $d' = 500$ m for ellipse pruning, $\alpha = 1.6$ for ratio-based pruning, $\gamma = \frac{C_1}{5}$ for cost-based pruning, and $(15, 2500, 4)$ for buckets. The multi-criteria route planning algorithm is implemented in JAVA 7. The results obtained are based on running the algorithm on a single core of a 2.4 GHz Intel Xeon E5-2665 processor of a Linux server. OpenStreetMap data is used to create the Prague cycleway graphs.

6.2 Evaluation Metrics

We consider two categories of evaluation metrics: *speed* and *quality*. We use the following metrics to measure the algorithm speed:

- Average runtime in ms for each origin-destination pair together with its standard deviation σ_{runtime}.
- Average speedup over the MLS algorithm in terms of algorithm runtime.

We use the following metrics to measure the quality of returned routes:

- Average distance $d_c(\Pi^*, \Pi)$ of the heuristic Pareto set Π from the optimal Pareto set Π^* in the cost space. Distance $d_c(\pi^*, \pi)$ between two routes π^* and π is measured as the Euclidean distance in the unit three-dimensional space of criteria values normalized to the $[0, 1]$ range.
$$d_c(\Pi^*, \Pi) := \frac{1}{|\Pi^*|} \sum_{\pi^* \in \Pi^*} \min_{\pi \in \Pi} d_c(\pi^*, \pi)$$
Intuitively, $d_c(\pi^*, \pi) = 0.1$ corresponds to a 6% difference in each criterion, assuming the difference to optimum is distributed equally across all three criteria.
- Average number of routes $|\Pi|$ in the Pareto set Π together with its standard deviation $\sigma_{|\Pi|}$.
- The percentage of Pareto routes $\Pi_{\%}$ in heuristic Pareto set Π that are equal to routes in the optimal Pareto set Π^*.

Table 2 Evaluation of the heuristic performance on three graphs Prague A, B, and C. Primary metrics are marked by bold column headings (runtime in ms and average distance $d_c(\Pi^*, \Pi)$). Non-dominated heuristic combinations with respect to speed and quality are denoted by bold font. Abbreviations used: Buckets → B., Ellipse → E., Ratio → R.

Heuristic	Speedup	**Runtime**	σ_{runtime}	$\|\Pi\|$	$\sigma_{\|\Pi\|}$	$\mathbf{d_c}$	$\Pi_{\%}$
MLS	-	3 586 263	4 390 939	1 351	1 304	-	100.0
HMLS+B.	875	**4 100**	3 335	37	32	**0.131**	60.9
HMLS+Cost	399	8 983	4 508	92	49	0.232	58.1
HMLS+R.	14	264 174	426 887	835	868	0.095	99.9
HMLS+R.+B.	2966	1 209	1 353	31	28	0.193	65.1
HMLS+R.+Cost	734	4 887	3 265	82	44	0.275	60.8
HMLS+E.	19	**184 734**	287 402	1 310	1 275	**0.008**	99.6
HMLS+E.+B.	6791	**528**	721	36	32	**0.136**	60.9
HMLS+E.+Cost	1732	2 070	1 976	91	49	0.235	58.5
HMLS+E.+R.	46	**77 468**	128 784	823	858	**0.098**	99.8
HMLS+E.+R.+B.	10308	**348**	461	31	28	**0.196**	65.1
HMLS+E.+R.+Cost	1921	1 866	1 902	82	44	0.276	61.0

6.3 Results

Table 2 summarizes the evaluation of the HMLS algorithm and its heuristics. The MLS algorithm is used as a baseline for the evaluation of the proposed heuristics and their combinations. Columns d_c and $\Pi_{\%}$ are calculated with respect to the optimal Pareto set Π^* returned by the MLS algorithm. The MLS algorithm returns optimal solutions (1351 routes in the Pareto set on average) at the expense of a prohibitively high runtime (one hour per one origin-destination pair on average).

As anticipated, all heuristic methods are significantly faster than the pure MLS algorithm. First, we have compared the methods using the two primary metrics in each category – the average runtime and the heuristic measured by the average distance $d_c(\Pi^*, \Pi)$ in the cost space. From the perspective of this two metrics, there are five non-dominated combinations of heuristics, cf. filled bars in Figure 3 and bold values in Table 2. In the following, we only discuss non-dominated combinations of heuristics.

The *HMLS+Ellipse* heuristic performs best in terms of the quality of the solution. It successfully prunes the search space with $d_c(\Pi^*, \Pi) = 0.008$. The average runtime of this heuristic is around three minutes. This heuristic is very good for combining with other heuristics, it offers one order of magnitude speedup over the MLS algorithm with a negligible quality loss (99.6% of the routes in the heuristic Pareto set Π are equal to the ones in the optimal Pareto set Π^*).

The *HMLS+Ellipse+Ratio* heuristic offers very good quality with $d_c(\Pi^*, \Pi) = 0.098$, the average runtime is around 80 seconds. The search space is pruned geographically by the ellipse pruning and the search is also terminated sooner by the ratio-based pruning method.

With only a small decrease of the solution quality to $d_c(\Pi^*, \Pi) = 0.131$, *HMLS+Buckets* heuristic offers a significant additional speedup in average runtime to approximately 4.1 seconds. This makes this heuristic (and also the two following ones) usable for real time applications, e.g., a web-based bicycle journey planner.

When the ellipse pruning method is combined with the *Buckets* heuristic, the average runtime of *HMLS+Ellipse+Buckets* is lowered to approximately 528 ms while keeping almost the same quality $d_c(\Pi^*, \Pi) = 0.136$.

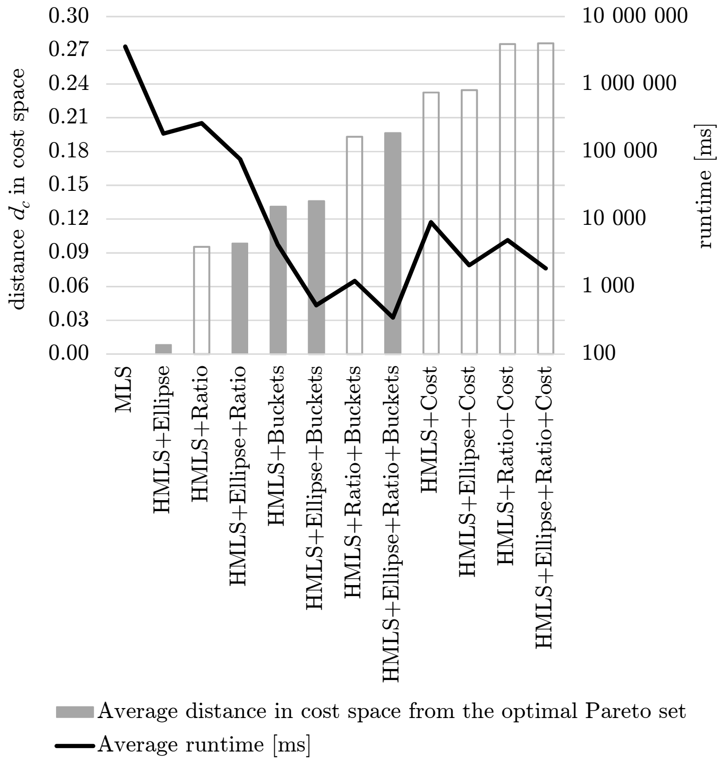

Figure 3 Speed and quality for the HMLS algorithm and all heuristic combinations sorted by the quality from the best (MLS on the left hand side) to the worst. Non-dominated heuristic combinations have grey filled in bars.

The last combination *HMLS+Ellipse+Ratio+Buckets* performs best in terms of average runtime which is approximately 350 ms, i.e., it has four orders of magnitude speedup over the pure MLS algorithm. The quality of this combination is reflected by higher $d_c(\Pi^*, \Pi) = 0.196$, still over 65% of the routes in the heuristic Pareto set Π are equal to the ones in the optimal Pareto set Π^*.

To provide a deeper insight in search runtimes, we show in Figure 4 how the runtime of the *HMLS+Ellipse+Buckets* heuristic depends on the direct origin-destination distance. Although the runtime increases with the origin-destination distance, the rate of increase slows down as the origin-destination distance grows. This behaviour was confirmed in our initial scale-up experiments that resulted in less than 10 second response times even for 20 times larger cycleway graph covering the whole city of Prague (approx. 200 km^2). Finally, in Figure 5 we illustrate the route distribution from the optimal Pareto set of routes in the physical space on an example of a route around a hilly area in Zizkov, Prague 3.

To summarize, we have evaluated 11 different combinations of heuristics from which 5 combinations dominated the others in terms of quality and speed. The heuristics offer significant *one to four orders of magnitude speedup* over the pure MLS algorithm in terms of average runtime. The speedup is achieved by lowering the number of iterations and also the number of dominance checks in each iteration. *HMLS+Ellipse* is the best heuristic in terms of quality of the produced Pareto set while *HMLS+Ellipse+Ratio+Buckets* is the best heuristic in terms of average runtime. Taking into the account the trade-off between the quality of a solution and the provided speedup, we consider *HMLS+Ellipse+Buckets* heuristic to have the best ratio between the quality and speed.

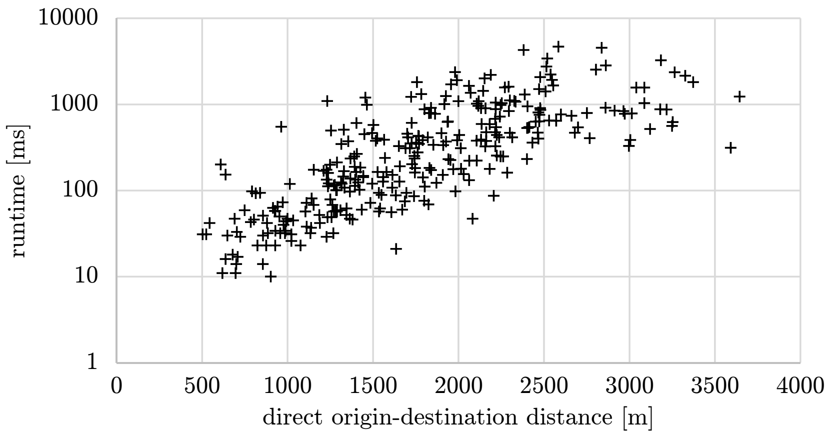

Figure 4 The runtime of *HMLS+Ellipse+Buckets* in milliseconds in dependency on the direct origin-destination distance.

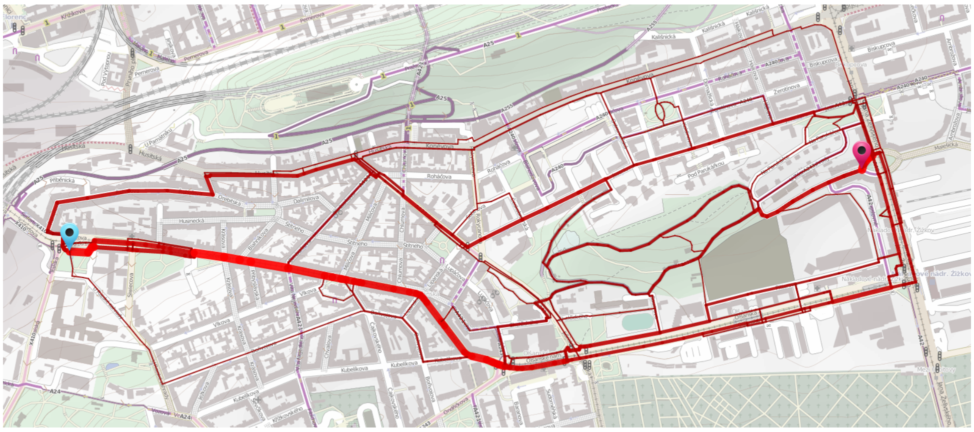

Figure 5 Distribution of 503 routes from the optimal Pareto set of routes. The more routes use a given cycleway network segment, the wider is the depicted line.

7 Conclusions

We have made bicycle routing that properly considers multiple realistic route choice criteria fast enough for practical, interactive use. We have achieved so by employing four heuristic speedup techniques for multi-criteria shortest path search. The speedup heuristics provide a variable trade-off between the search time and the completeness and quality of the suggested routes and they enable fast response times without severely compromising the quality of the results.

The multi-criteria search produces often large Pareto sets with many similar routes. As a future work, we plan to provide a filtering method (e.g., based on our initial clustering method [24]) that would extract several representative routes from a potentially very large set of Pareto routes. Furthermore, we plan to extend the underlying cycleway graph model to consider additional features such as detailed junction models with traffic lights and penalisation of turns.

Acknowledgements. Supported by the European social fund within the framework of realising the project "Support of inter-sectoral mobility and quality enhancement of research teams at Czech Technical University in Prague" (grant no. CZ.1.07/2.3.00/30.0034), period of the project's realisation is 1 Dec 2012 – 30 Jun 2015. Supported by the European Union Seventh Framework Programme (grant agreement no. 609023) and by the Czech Technical University (grant no. SGS13/210/OHK3/3T/13). Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005), is greatly appreciated.

References

1 H. Bast, D. Delling, A. Goldberg, M. Muller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. Werneck. Route Planning in Transportation Networks. Technical report, Microsoft Research, 2014.

2 Hannah Bast, Mirko Brodesser, and Sabine Storandt. Result Diversity for Multi-Modal Route Planning. In Daniele Frigioni and Sebastian Stiller, editors, *13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 33 of *OpenAccess Series in Informatics (OASIcs)*, pages 123–136, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

3 J. Broach, J. Dill, and J. Gliebe. Where do cyclists ride? A route choice model developed with revealed preference GPS data. *Transportation Research Part A: Policy and Practice*, 46(10):1730 – 1740, 2012.

4 D. W. Corne. The good of the many outweighs the good of the one: evolutionary multi-objective optimization. *IEEE Connections Newsletter*, pages 9–13, 2003.

5 B. C. Dean. Continuous-time dynamic shortest path algorithms. Master's thesis, Massachusetts Institute of Technology, 1999.

6 D. Delling, J. Dibbelt, T. Pajor, D. Wagner, and R. F. Werneck. Computing and Evaluating Multimodal Journeys. Technical Report 2012-20, Faculty of Informatics, Karlsruhe Institut of Technology, 2012.

7 D. Delling, J. Dibbelt, T. Pajor, D. Wagner, and R. F. Werneck. Computing multimodal journeys in practice. In *SEA*, pages 260–271, 2013.

8 D. Delling and D. Wagner. Pareto paths with sharc. In *Proceedings of the 8th International Symposium on Experimental Algorithms, volume 5526 of LNCS*, pages 125–136. Springer, 2009.

9 D. Delling and D. Wagner. Time-dependent route planning. In *Robust and Online Large-Scale Optimization*, pages 207–230. Springer, 2009.

10 C. Dora and M. Phillips. *Transport, environment and health.* WHO Regional Office for Europe, Copenhagen, 2000.

11 V. Filler. (AUTO*MAT) Private communication, 2013. Why cycle route planners are important for cyclists.

12 L. Han, H. Wang, and W. Mackey Jr. Finding shortest paths under time-bandwidth constraints by using elliptical minimal search area. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1977:225–233, 2006.

13 D. Herlihy. *Bicycle: the history.* Yale University Press, 2004.

14 H. H. Hochmair and J. Fu. Web Based Bicycle Trip Planning for Broward County, Florida. In *ESRI User Conference*, 2009.

15 J. Hrncir, Q. Song, P. Zilecky, M. Nemet, and M. Jakob. Bicycle route planning with route choice preferences. In *Prestigious Applications of Artificial Intelligence (PAIS)*, 2014.

16 P. L. Jacobsen. Safety in numbers: more walkers and bicyclists, safer walking and bicycling. *Injury Prevention*, 9(3):205–209, 2003.

17 E. Machuca and L. Mandow. Multiobjective heuristic search in road maps. *Expert Systems with Applications*, 39(7):6435–6445, 2012.

18 L. Mandow and J. De La Cruz. Multiobjective A* search with consistent heuristics. *J. ACM*, 57(5), June 2008.

19 E. Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236 – 245, 1984.

20 M. Muller-Hannemann and M. Schnee. Finding all attractive train connections by multi-criteria pareto search. In *ATMOS*, pages 246–263, 2004.

21 M. Müller-Hannemann and K. Weihe. On the cardinality of the pareto set in bicriteria shortest path problems. *Annals of Operations Research*, 147(1):269–286, 2006.

22 P. Perny and O. Spanjaard. Near admissible algorithms for multiobjective search. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 490–494, Amsterdam, The Netherlands, 2008. IOS Press.

23 G. Sauvanet and E. Neron. Search for the best compromise solution on multiobjective shortest path problem. *Electronic Notes in Discrete Mathematics*, 36:615–622, 2010.

24 Q. Song, P. Zilecky, M. Jakob, and J. Hrncir. Exploring pareto routes in multi-criteria urban bicycle routing. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1781–1787, Oct 2014.

25 B. S. Stewart and Ch. C. White. Multiobjective A*. *Journal of the ACM (JACM)*, 38(4):775–814, 1991.

26 J. G. Su, M. Winters, M. Nunes, and M. Brauer. Designing a route planner to facilitate and promote cycling in Metro Vancouver, Canada. *Transportation Research Part A: Policy and Practice*, 44(7):495–505, 2010.

27 R. J. Turverey, D. D. Cheng, O. N. Blair, J. T. Roth, G. M. Lamp, and R. Cogill. Charlottesville bike route planner. In *Systems and Information Engineering Design Symposium (SIEDS)*, 2010.

28 M. Winters, G. Davidson, D. Kao, and K. Teschke. Motivators and deterrents of bicycling: comparing influences on decisions to ride. *Transportation*, 38(1):153–168, 2011.