Optimal Bicycle Routes with Few Signal Stops

Ekkehard Köhler
Institute of Mathematics, Brandenburg University of Technology, Cottbus, Germany

Markus Rogge
Institute of Mathematics, Brandenburg University of Technology, Cottbus, Germany

Robert Scheffler
Institute of Mathematics, Brandenburg University of Technology, Cottbus, Germany

Martin Strehler
Department of Mathematics, Westsächsische Hochschule Zwickau, Germany

Abstract

With the increasing popularity of cycling as a mode of transportation, there is a growing need for efficient routing algorithms that consider the specific requirements of cyclists. This paper studies the optimization of bicycle routes while minimizing the number of stops at traffic signals. In particular, we consider three different types of stopping strategies and three types of routes, namely paths, trails, and walks. We present hardness results as well as a pseudo-polynomial algorithm for the problem of computing an optimal route with respect to a pre-defined stop bound.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Mathematics of computing → Paths and connectivity problems; Applied computing → Transportation

Keywords and phrases Constrained shortest path, traffic signals, bicycle routes

Digital Object Identifier 10.4230/OASIcs.ATMOS.2023.1

1 Motivation

Cities around the world are increasingly recognizing the importance of promoting sustainable and efficient transportation options, including bicycle traffic. Cycling has been shown to have numerous health and ecological benefits, including reducing air pollution, increasing physical activity, and reducing greenhouse gas emissions. Consequently, there has been a growing interest in exploring alternative approaches to optimize bicycle traffic in cities and many factors such as traffic volume, road surface quality, and the availability of bike lanes can help making cycling a more efficient and enjoyable mode of transportation.

Research has shown that the travel time and the length of the route are not the only factors that influence the route choice of cyclists. Other factors such as the number of stops are also crucial in determining the attractiveness of a given route. Often, cyclists prefer routes with fewer stops and more opportunities for continuous movement [15], because frequent stops and starts can be physically demanding, reduce overall speed, and increase the likelihood of accidents.

To reduce the number of stops for cyclists, there are two main options. The first option is to optimize traffic signals to create a more efficient flow of bicycle traffic. The second option is to offer cyclist-dependent routes that are specifically designed to match the speed of the cyclists and which, using information about the signal coordination, arrive at green traffic lights most of the time.

Implementing the first of these two options can be very challenging compared to optimizing traffic lights for cars. Motorized vehicles generally operate within a narrow speed range and tend to form platoons, which can be exploited in signal coordination [8, 9] and is used in signal control systems, such as TRANSYT [12] or SCOOT [6]. In contrast, cyclists' speeds can vary

© Ekkehard Köhler, Markus Rogge, Robert Scheffler, and Martin Strehler;
licensed under Creative Commons License CC-BY 4.0
23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023).
Editors: Daniele Frigioni and Philine Schiewe; Article No. 1; pp. 1:1–1:14
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

by a factor of three or more and cyclists are much less likely to form platoons [18]. Especially when optimizing coordination for both bicycles and cars simultaneously, this causes similar difficulties as, for example, the integration of public transit priority into coordination [14].

In contrast, offering a route optimized for cyclists, along with real-time traffic updates and available through a smartphone application, can significantly enhance the cycling experience and encourage more people to consider cycling as a mode of transportation. Consequently, here we will study the underlying algorithmic problems of finding routes with few stops.

Related Literature. The problem of finding shortest routes with few stops has similarities to dynamic shortest paths with time-dependent travel times, to shortest paths with time windows, as well as to constrained shortest paths.

Time-dependent travel times are often used in traffic literature to model the influence of slow-moving traffic and congestion. In particular, the first in-first out property (FIFO) must be taken into account. FIFO means that later entry into an edge also means later arrival at the end of an edge, so overtaking is not possible.

If FIFO holds, a shortest path can be found with Dijkstra-like approaches [3]. In non-FIFO settings, Orda and Rom [10] observed that there may not even exist a finite solution, i.e., a route with finitely many links, if waiting at vertices is not allowed. Recently, Zeitz [19] elaborated these results by showing that this problem is strongly NP-hard even for piecewise linear delay functions represented by a sequence of breakpoints with integer coordinates. However, waiting in this context always means voluntary waiting to profit from a better transit time in the future, whereas mandatory waiting, e.g, at red signals, is directly included in the travel time function.

A generalized version with separate delay and cost functions including costs for waiting was investigated by Orda and Rom [11]. The authors show that a suitable choice of these functions leads to the situation that no finite optimal solution exists, but they also discuss various conditions that guarantee the existence of finite solutions.

Traffic signals induce a special kind of delay, where FIFO usually holds. Consequently, in a network with periodic traffic signals, shortest paths with respect to travel time can be found in polynomial time as was shown by Ahuja et al. [1]. In the same paper, the authors also consider minimum cost paths, in particular, they use penalties to give more weight to waiting. Here, the problem of finding a minimum cost path becomes NP-hard in general.

Another application of routing with periodic time windows was investigated by Kleff et al. [7]. These authors compute the set of Pareto-optimal solutions for the route planning problem with temporary driving bans, like driving bans on trucks on Sundays, and rated parking areas, i.e., different locations cause different costs for waiting.

In our approach, we are going to count stops separately, that is, stops can be seen as a bounded resource. Such constrained shortest paths are weakly NP-complete in general, see, e.g., [5]. However, we use a rather special resource consumption function here which only applies during periodic time intervals. We will see that this may cause the optimal route to contain cycles. Similar effects can be observed in energy-efficient routing with variable resource constraints [17].

Our Contribution. In this paper, we study the routing with periodic time windows induced by traffic signals for different route types (namely paths, walks, and trails), different cyclist types with respect to waiting behavior, and for bounded or unlimited number of stops, respectively. In Section 2, we present the notation and basic properties. Complexity results are discussed in Section 3 and algorithmic results are presented in Section 4. Note that some results of this paper are based on the Master's thesis of Markus Rogge [13].

2 Basic Model and Properties

In this section, we will fix the notation, introduce the model, and show some of its basic properties.

2.1 Basic Model

We start with an underlying graph with vertex set V and edge set E that is supposed to represent our road network. Without loss of generality, we use directed edges here, since every bidirectional road can be modeled by a forward and a backward arc. Each edge also has an integral transit time which is given by the function $\tau : E \to \mathbb{N}_0$. Furthermore, for a realistic modeling of practical instances, we use expanded intersections. That is, an intersection is not only represented by a single vertex, but there are vertices for each entry and exit, as well as arcs for each permissible turning direction. A "standard" intersection is shown in Figure 1.

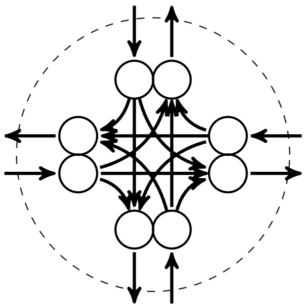

Figure 1 Standard approach of handling different turning directions at a single intersection with separate arcs. Here, two roads are crossing each other. Thus, there are four entry points and four exit points at the intersection. If, for example, left turning is not allowed, then the corresponding arcs can be deleted.

Some edges $E' \subseteq E$ of our road network are equipped with traffic signals. Usually, these should be interior edges of the expanded intersections (see Figure 1). To keep it simple, we assume that all traffic signals follow a periodic switching regime with a common cycle time and that all signals have exactly one green phase and one red phase per cycle[1]. Taking all these parts together, this yields a signalized network.

▶ **Definition 1.** *A* signalized network $N = (V, E, E', T, \tau, \gamma, \lambda)$ *consists of*

- *a vertex set V,*
- *a set of directed edges E,*
- *a subset $E' \subseteq E$ of edges with signals,*
- *a common cycle time $T \in \mathbb{N} = \{1, 2, 3, \dots\}$,*
- *a transit time function $\tau : E \to \mathbb{N}_0 = \{0, 1, 2, \dots\}$,*
- *a function $\gamma : E' \to \{0, \dots, T-1\}$ describing the start time of a green phase, and*
- *a function $\lambda : E' \to \{1, \dots, T-1\}$ describing the length of the green phase.*

[1] More sophisticated settings can also be modelled with these assumptions. If there is no common cycle time, then one could consider the least common multiple of all cycle times. Several green phases within one cycle can be modelled via parallel arcs where every of those arcs represents one of the green phases.

When moving through the network, a red traffic light on edge $e \in E'$ keeps us from entering this edge. To determine whether a signal is red or green at a given point in time, we define the function $\texttt{green} : E' \times \mathbb{N}_0 \to \{0, 1\}$ that returns 1 if and only if the traffic light of the given edge is green at the given time step. More precisely, $\texttt{green}$ is defined as follows:

$$\texttt{green}(e,t) = \begin{cases} 1, & \text{if } kT + \gamma(e) \leq t < kT + \gamma(e) + \lambda(e) \text{ for some } k \in \mathbb{N}_0 \\ 0, & \text{otherwise} \end{cases}$$

Note that the function $\texttt{green}$ is periodic in the argument t with period length T.

Now, we are looking for a route for a cyclist starting at an origin vertex $s \in V$, ending in a destination vertex $d \in V$. In general, we would like to allow to visit vertices or edges more than once. Therefore, we distinguish walks, trails, and paths. A *walk* is an ordered list $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$ of vertices and edges such that for any $i \in \{1, \ldots, k\}$ edge $e_i = (v_{i-1}, v_i)$. A *trail* is a walk where every edge is present at most once. A *path* is a trail where every vertex is present at most once.

In difference to the "normal" shortest path problem, in our setting it is possible to wait at certain vertices. Therefore, we need the following extension of the definitions of walks, trails, and paths. A *timed walk (trail, path)* $\mathcal{P} = (P, \pi)$ consists of a walk (trail, path) $P = (v_0, e_1, \ldots, e_k, v_k)$ and a time function π that assigns to every edge e_i of P an entering time from $\mathbb{N}_0$ such that the following conditions hold:

- for every edge e_i it holds that $\pi(e_i) \geq \pi(e_{i-1}) + \tau(e_{i-1})$ and
- for every edge $e_i \in E'$ it holds that $\texttt{green}(e_i, \pi(e_i)) = 1$.

The first condition ensures that an edge is not entered before the previous edge has been traversed. Nevertheless, waiting is possible here. The second condition ensures that the cyclists only enter signalized edges if the traffic signal is green. Note that edges can appear more than once in $\mathcal{P}$ if $\mathcal{P}$ is not a trail. Every appearance of an edge is treated separately and is assigned its own π-value.

We always start at time $t = 0$. The *arrival time* $\alpha(\mathcal{P})$ of a timed walk (trail, path) $\mathcal{P} = (P, \pi)$ is the time when the cyclist arrives at the last vertex of P, i.e., if e_k is the last edge of P, then $\alpha(\mathcal{P}) = \pi(e_k) + \tau(e_k)$.

We are especially interested in the number of stops on our timed path $\mathcal{P}$. A *stop* on edge e_i of $\mathcal{P}$ occurs if either $i = 1$ and $\pi(e_i) > 0$ or $i > 1$ and $\pi(e_i) > \pi(e_{i-1}) + \tau(e_{i-1})$. This in particular means that we count waiting on the first edge as a stop no matter whether there is a red light or not. We define $\sigma(\mathcal{P})$ as the number of stops of the timed walk (trail, path) $\mathcal{P}$.

Moreover, we consider three different types of cyclists:

- *Impatient cyclists* do not stop at green signals. When waiting at a red signal, they start as soon as the signal turns green.
- *Predictive cyclists* also do not stop at green signals, but they are allowed to wait at a green light if they arrived there during the red phase.
- *Relaxed cyclists* are allowed to stop wherever they want. In particular, they are allowed to stop at green signals and on edges $E \setminus E'$.

Figure 2 visualizes the differences of the three cyclist types. Note that in reality most cyclists will behave like the impatient cyclists. However, we will show later (Lemma 5) that predictive cyclists have an advantage over impatient cyclists in certain situations. On the other hand, relaxed cyclists will not perform better than predictive cyclists.

We will study two different routing problems in this paper.

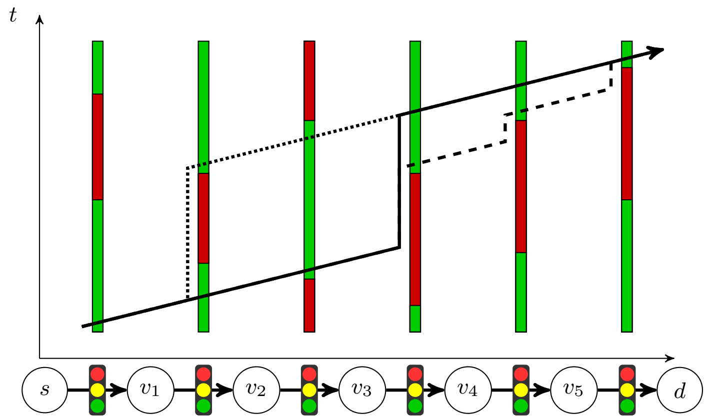

Figure 2 Time-space diagram of the three cyclist types impatient (dashed), predictive (solid), and relaxed (dotted). The impatient cyclist has to endure three stops. The predictive cyclist waits extra time at the first stop at the fourth traffic signal on edge (v_3, v_4) so that the destination can be reached without another stop. The relaxed cyclist is already waiting at a green traffic signal.

▶ **Problem 1** (Quickest path (trail, walk)).
Input: *A signalized network* $N = (V, E, E', T, \tau, \gamma, \lambda)$*, source* $s \in V$*, destination* $d \in V$*.*
Task: *Compute a timed path (trail, walk)* $\mathcal{P}$ *starting in* s *and ending in* d *such that the arrival time* $\alpha(\mathcal{P})$ *is minimal.*

▶ **Problem 2** (Quickest path (trail, walk) with few stops).
Input: *A signalized network* $N = (V, E, E', T, \tau, \gamma, \lambda)$*, source* $s \in V$*, destination* $d \in V$*, stop bound* $K \in \mathbb{N}_0$*.*
Task: *Compute a timed path (trail, walk)* $\mathcal{P}$ *starting in* s *and ending in* d *with* $\sigma(\mathcal{P}) \leq K$ *such that the arrival time* $\alpha(\mathcal{P})$ *is minimal.*

For both problems, we have to consider three types of routes and three types of cyclists, which initially results in 18 different problems that have to be studied.

2.2 Basic Properties

We continue this section with discussing some basic properties of the model. First, we present an example to justify the consideration of trails and walks.

▶ **Example 2.** Consider the network in Figure 3. All edges have transit time $\tau \equiv 1$. There is a single traffic signal at edge $e' = (v, d)$. For a suitable choice of parameters, it can be advantageous to travel through the cycle and to use $e = (u, v)$ more than once. If, e.g., $T = 20$, $\gamma(e') = 10$, and $\lambda(e') = 5$, then the s-d-path P using the cycle three times (that is, e is traversed four times) yields an arrival time $\alpha(\mathcal{P}) = 12$ at vertex d. This is the optimal solution for $K = 0$. Yet, it is possible to arrive at d at time 11, but this implies a stop somewhere in the network.

The classic shortest path problem fulfills *subpath optimality*, i.e., every subpath of an optimal s-d-path ending in some intermediate vertex v is also an optimal s-v-path. This crucial property is exploited by many shortest path algorithms such as Dijkstra's algorithm. We can use Example 2 to show that this property does not hold in our setting.

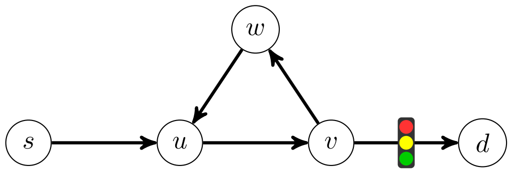

Figure 3 Simple network with $\tau \equiv 1$ for all edges and a single traffic signal on edge $e' = (v, d)$. Vertex v can be reached at time $t = 2$, but if the signal is not green at this time, it can be advantageous to use the cycle and, thus, edge $e = (u, v)$ again to avoid stopping.

▶ **Observation 3.** *There is no guaranteed subpath optimality for quickest paths (walks, trails) with few stops. In particular, it can be advantageous to arrive later at a particular intermediate vertex.*

Proof. We again consider the network in Figure 3 and we add an edge e'' from s to v with $\tau(e'') = 10$. Although, v can be reached at time $t = 2$ via u, this always implies a stop or a cycle in the route. Thus, using e'' and arriving at v no sooner than $t = 10$ is the optimal path without stopping. ◀

The previous observation implies that we have to deal with a much richer combinatorial variety if we want to solve the problem algorithmically. However, taking a closer look at the above mentioned 18 subproblems, we observe that some of these cases coincide.

▶ **Observation 4.** *The time function π of a timed path (trail, walk) for the impatient cyclists is completely determined by its edges. Contrary, every timed path (trail, walk) is feasible for the relaxed cyclists.*

The following result shows that we can always adjust a given route such that it is feasible for predictive cyclists without increasing the number of stops or the arrival time and without changing the used edges.

▶ **Lemma 5.** *Let $N = (V, E, E', T, \tau, \gamma, \lambda)$ be a signalized network and $s, d \in V$. For any timed s-d-path (trail, walk) $\mathcal{P} = (P, \pi)$ there is a timed s-d-path (trail, walk) $\mathcal{P}' = (P, \pi')$ with $\alpha(\mathcal{P}') \leq \alpha(\mathcal{P})$ and $\sigma(\mathcal{P}') \leq \sigma(\mathcal{P})$ that is feasible for the predictive cyclist.*

Proof. We construct the path $\mathcal{P}'$ using the same edges as in $\mathcal{P}$, i.e., we only adjust the time function π' of $\mathcal{P}'$. Due to Observation 4, $\mathcal{P}$ is feasible for the relaxed cyclists, but there may be stops that are not allowed for the predictive cyclists.

Let e_i be the first edge where $\mathcal{P}$ stops, but predictive cyclists are not allowed to stop at e_i. We change the time function π' as follows. $\mathcal{P}'$ continues directly, i.e., if $i > 1$, then $\pi'(e_i) = \pi'(e_{i-1}) + \tau(e_{i-1})$ and if $i = 1$, then $\pi'(e_i) = 0$. This also implies that up to this point $\mathcal{P}'$ has fewer stops than $\mathcal{P}$ since the stop at e_i was removed. Also at subsequent green and non-signalized edges, $\mathcal{P}'$ does not stop according to the rule for the predictive cyclists. Only when $\mathcal{P}'$ arrives at a red signal at edge e_j, $\mathcal{P}'$ uses this stop to wait for $\mathcal{P}$, that is, $\pi'(e_j) = \pi(e_j)$. This adds one more stop to $\mathcal{P}'$, but still $\mathcal{P}'$ has at most as many stops as $\mathcal{P}$. Figure 2 shows this construction for a single stop in a time-space diagram where the path $\mathcal{P}'$ of the predictive cyclist is visualized by the solid line and the path $\mathcal{P}$ of the relaxed cyclist is visualized by the dotted line.

We continue with this procedure for all further occurrences of stops of $\mathcal{P}'$ that are invalid for the predictive cyclists. In this way, $\mathcal{P}'$ arrives at the destination d not later than $\mathcal{P}$ and has at most as many stops as $\mathcal{P}$. ◀

As a consequence we can refrain from considering the relaxed cyclists separately, as for each route of the relaxed cyclists there is a route of the predictive cyclists with equal or better arrival time and number of stops.

▶ **Lemma 6.** *If there is no bound on the number of stops, then the set of quickest trails/walks for a given s-d-pair always contains a timed s-d-path that is feasible for the impatient cyclists.*

Proof. We only need to discuss this claim for the impatient cyclists, since routes for the impatient cyclists are also optimal for the predictive/relaxed cyclists if the number of stops does not matter, as it follows directly from the definition, that the predictive and the relaxed cyclists never overtake the impatient cyclists on the same route.

Suppose an optimal timed route $\mathcal{P} = (P, \pi)$ contains a vertex at least twice, for example $v_i, v_j \in P$ with $v_i = v_j$ and $i < j$. This means there is a cycle within the route. We can delete this cycle from the route, i.e., delete all vertices from v_{i+1} to v_j and the corresponding edges. To again obtain a proper timed route for the impatient cyclist, we have to adjust the entering time of (v_i, v_{j+1}) and all subsequent edges in π. We simply choose the earliest possible feasible time. Note that due to the optimality of the original route, there is a vertex v_k, $k > j$, from which the time function no longer needs to be adjusted.

Repeating this procedure iteratively for all cycles yields a feasible path for the impatient cyclists that has the same arrival time as $\mathcal{P}$. ◀

This result also implies that we can bound the maximal number of stops of a quickest walk.

▶ **Proposition 7.** *In a signalized network with $n = |V|$ edges, there exists a quickest path (trail, walk) with at most $n - 1$ stops.*

Proof. By Lemma 6, there is a quickest path from s to d that is feasible for the impatient cyclists and, thus, also for the predictive and the relaxed cyclists. This path contains at most n vertices and, hence, $n - 1$ edges. Therefore, it also has at most $n - 1$ stops. ◀

In consequence, the problem of the quickest path (trail, walk) with few stops is only interesting for $K < n$.

3 Hardness Results

In this section we will prove that Problem 2 is NP-hard for path, trail and walk and any type of cyclists. To this end, we consider the following problem.

▶ **Problem 3** (x-y-Hamiltonian Path).
Input: *A directed graph $G = (V, E)$, two vertices $x, y \in V$*
Task: *Decide whether there is a Hamiltonian path in G, i.e., a path containing all vertices of V, that starts in x and ends in y.*

This problem is NP-complete [4]. We now prove that the Quickest Path and the Quickest Trail Problem with few stops are strongly NP-hard, i.e., there is no pseudo-polynomial algorithm unless P = NP.

▶ **Theorem 8.** *The Quickest Path Problem with few stops and the Quickest Trail Problem with few stops (Problem 2) are strongly* NP*-hard for all three types of cyclists and every fixed number of stops $K \geq 0$.*

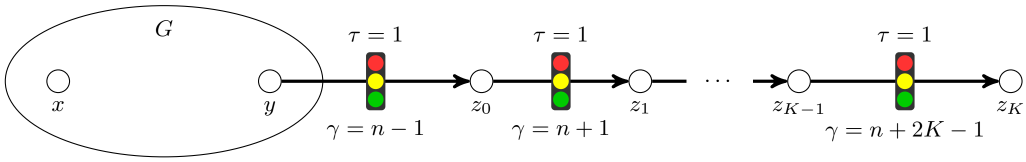

Figure 4 The construction of the proof of Theorem 8.

Proof. We start with the Quickest Path Problem. We reduce the x-y-Hamiltonian Path Problem to that problem. Let $G = (\tilde{V}, \tilde{E})$ and $x, y \in \tilde{V}$ be an instance of the x-y-Hamiltonian Path Problem. Let $n = |\tilde{V}|$ and let $K \geq 0$ be an arbitrary fixed stop bound. We define the signalized network $N = (V, E, E', T, \tau, \gamma, \lambda)$ as follows (see Figure 4 for an illustration). The vertex set V is equal to $\tilde{V} \cup \{z_0, \ldots, z_K\}$. The set of edges with signals E' is equal to $\{(y, z_0)\} \cup \{(z_i, z_{i+1}) \mid 0 \leq i \leq K-1\}$. The edge set E is equal to $\tilde{E} \cup E'$. The cycle time T is set to $n + 2K$. For all edges $e \in E$ we set $\tau(e) = 1$. We set the starting time of the green phase $\gamma((y, z_0)) = n - 1$ and the green length to $\lambda((y, z_0)) = 1$. Furthermore, we set $\gamma((z_i, z_{i+1})) = n + 2i + 1$ and $\lambda((z_i, z_{i+1})) = 1$ for all $i \in \{0, \ldots, K-1\}$.

We claim that there is a timed path $\mathcal{P}$ starting in x and ending in z_K with no more than K stops and an arrival time less than or equal to $n + 2K$ if and only if there is an Hamiltonian path from x to y in G.

First assume that we have a Hamiltonian path from x to y in G. Following this path in N, the cyclists arrive at y without any stop at time step $n - 1$. Since $\gamma((y, z_0)) = n - 1$, they can enter edge (y, z_0) without any stop and arrive at z_0 at time step n. Since the green phase of the edge (z_0, z_1) starts at time step $n + 1$, the cyclists have to stop at z_0 and arrive at z_1 at time step $n + 2$. Repeating this for all i, we see that the cyclists always arrive at z_i at time step $n + 2i$ and have to wait one time step for green. Thus, finally the cyclists arrive at z_K at time step $n + 2K$ having stopped exactly K times.

Now assume that there is a timed path $\mathcal{P} = (P, \pi)$ in N starting in x and ending in z_K with no more than K stops and an arrival time less than or equal to $n + 2K$. This path has to enter the edge (y, z_0) at time step $n - 1$ at the latest since otherwise the next green phase will start only in the next time frame at time step $2n + 2K - 1$. Note that before time step $n - 1$, the edge (y, z_0) is not green. Hence, the entering time $\pi((y, z_0))$ is exactly $n - 1$. Using the same arguments as above, we see that the cyclists have to stop at every vertex z_i with $0 \leq i < K$. Hence, the cyclists do not stop at any vertex before z_0. This implies that the subpath of P between x and y must contain $n - 1$ edges and, thus, this subpath forms a Hamiltonian path of G between x and y.

We now reduce the Quickest Path Problem with few stops to the Quickest Trail Problem with few stops. To this end, we replace every vertex v in the network N by two vertices v_{in} and v_{out}. All incoming edges of v in N now ends in v_{in} and all outgoing edges start now in v_{out}. Furthermore, we add the edge (v_{in}, v_{out}) with transit time 0. We call the resulting network N'. It is easy to observe that there is a one-to-one mapping from the timed paths in the original network to timed paths in N' with the same arrival time and the same number of stops. Furthermore, every trail in N' is a path. Therefore, solving the Quickest Trail Problem with few stops on N' solves the Quickest Path Problem with few stops on N.[2]

[2] Note that one can also prove the hardness of the Quickest Trail Problem if one forbids edges with transit time 0. The idea is to prove that the x-y-Hamiltonian Path Problem stays NP-hard even if the input is a directed graph that was created by replacing every vertex v by two vertices v_{in} and v_{out} as described above.

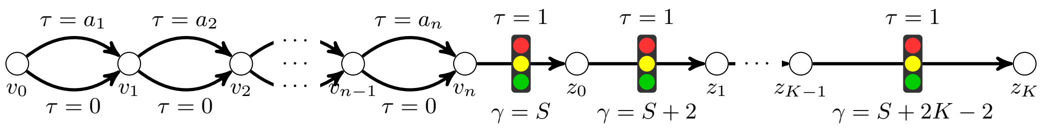

Figure 5 The construction of the proof of Theorem 9.

Note that in both proofs the size of the used integers is polynomial in n (since K is a fixed constant). Therefore, any pseudo-polynomial algorithm for the Quickest Path (Trail) Problem would be polynomial in n on the constructed instances and, thus, both problems are strongly NP-hard. ◀

The idea of the proof of Theorem 8 does not work for walks since one cannot prevent that cycles are used to achieve the correct arrival time at the first traffic light. Nevertheless, we can show that the Quickest Walk Problem with few steps is NP-hard using the following well-known NP-hard problem.

▶ **Problem 4** (Partition).
Input: *A set* $A = \{a_1, \dots, a_n\} \subsetneq \mathbb{N}$.
Task: *Decide whether there is a subset* $A' \subseteq A$ *such that* $\sum_{a_i \in A'} a_i = \frac{1}{2} \sum_{a_i \in A} a_i$.

Note that the Partition problem is known to be weakly NP-hard, i.e., there is a pseudo-polynomial algorithm for that problem [4].

▶ **Theorem 9.** *The Quickest Walk Problem with few stops is* NP*-hard for all three types of cyclists and every fixed number of stops* $K \geq 0$.

Proof. We reduce the Partition problem to our problem. Let $A = \{a_1, \dots, a_n\} \subsetneq \mathbb{N}$ be a an instance of the Partition problem. Let $S = \frac{1}{2} \sum_i a_i$. Let $K \geq 0$ be an arbitrary stop bound. We define the signalized network $N = (V, E, E', T, \tau, \gamma, \lambda)$ as follows (see Figure 5 for an illustration). Let $V = \{v_0, \dots, v_n, z_0, \dots, z_K\}$. For all $i \in \{1, \dots, n\}$, we have two edges e_i^1 and e_i^2 in E from v_{i-1} to v_i where $\tau(e_i^1) = a_i$ and $\tau(e_i^2) = 0$. Furthermore, for all $i \in \{1, \dots, K\}$, we have an edge $e_i^z = (z_{i-1}, z_i)$ in E and E' with $\tau(e_i^z) = 1$, $\gamma(e_i^z) = S + 2i$ and $\lambda(e_i^z) = 1$. Similar, we have an edge (v_n, z_0) with $\tau((v_n, z_0)) = 1$, $\gamma((v_n, z_0)) = S$ and $\lambda((v_n, z_0)) = 1$. We choose T to be $S + 2K$. Note that every walk in N is a path since the network is acyclic.

We claim that there is a subset $A' \subseteq A$ such that $\sum_{a_i \in A'} a_i = S$ if and only if there is a timed path from v_0 to z_K with arrival time at most $S + 2K - 1$ and at most K stops. First assume that there is a subset $A' \subseteq A$ such that $\sum_{a_i \in A'} a_i = S$. We choose the following path P from v_0 to z_K. If $a_i \in A'$, then we choose the edge e_i^1, else we choose the edge e_i^2. Furthermore, we choose all the edges e_i^z with $1 \leq i \leq K$. We can choose the time function π of the path in such a way that the cyclists arrive at v_n at time step S without any stop because $\sum_{a_i \in A'} a_i = S$. Since the edge (v_n, z_0) is green at this time step, the cyclists can enter it and arrive at z_1 at time step $S + 1$. Here, the cyclists have to wait one time step for green. Repeating this argument, the cyclists arrive at z_i at time step $S + 2i - 1$ and have to wait there for one time step. Overall, the cyclists arrive at z_K at time step $S + 2K - 1$ with K stops.

Now assume that there is a timed path $\mathcal{P} = (P, \pi)$ from v_0 to z_K with arrival time at most $S + 2K - 1$ and at most K stops. As described above, it follows directly from the construction that this path has to stop at edge (z_i, z_{i+1}) for all $i \in \{0, \dots, K-1\}$. Hence, the path does not stop neither at an edge between vertices v_i and v_{i+1} nor at the edge (v_n, z_0). This implies that the path arrives at v_n at time step S. We define the set A' as follows: $A' := \{a_i \mid e_i^1 \in P\}$. By the observation before, it must hold that $\sum_{a_i \in A'} a_i = S$. ◀

Chen and Yang [2] presented an algorithm for which they claim that, given a signalized network, it finds a quickest path with a given maximal number of stops K in time $\mathcal{O}(Kn^3)$, where n is the number of vertices in the network. The authors do not specify what they mean with the term "path". Nevertheless, as we have seen in Theorem 8, all three options for routes (path, walk, trail) are NP-hard. This either implies that P = NP or, more likely, there is a flaw in the algorithm of Chen and Yang. In fact, we will show in the next section, where Chen and Yang's algorithm fails. Furthermore, we present an alternative approach that has pseudo-polynomial running time and computes the quickest walk with few stops.

4 Algorithmic Results

Ahuja et al. [1] showed that signalized networks are *FIFO graphs*, i.e., arriving earlier at a particular vertex can never result in arriving later at one of the following vertices. Using a result by Dreyfus [3] from the 1960s, they showed that this result implies a polynomial-time algorithm for quickest paths in signalized networks, i.e., Problem 1.

As mentioned in the last section, Chen and Yang [2] presented an algorithm that, as they claim, solves Problem 2. This algorithm uses a labeling approach similar to Dijkstra's algorithm for shortest paths. While Dijkstra's algorithm holds at most one label for every vertex (the label with the current best arrival time at that vertex), Chen and Yang's algorithm holds for every vertex v and for every number k of stops the walk from s to v with at most k stops and the best arrival time of all s-v-walks found so far. Thus, if a path arrives at a vertex v and then via a cycle arrives at v again, then the arrival time and the number of stops are not better than at the first arrival at v. This implies that this algorithm will never use cycles. However, as we have seen in Example 2, it can be necessary to use cycles to stay within a given stop bound. Furthermore, even if we forbid cycles, i.e., we restrict to paths, Chen and Yang's algorithm fails to find the quickest path with a given number of stops. This is due to the fact that subpath optimality does not hold in this setting as we have seen in Observation 3. It is easy to see that Chen and Yang's algorithm already fails to solve the example given in the proof of that observation.

To overcome this problem, we present an alternative approach that not only takes these non-FIFO behaviour and possible cycles into account but also distinguishes between the different types of cyclists that we have introduced in Section 2. Due to Lemma 5, we do not need to consider the relaxed cyclists.

We will use the following terminology. The *absolute time* refers to the time the cyclist has used so far in total. The *relative time* is the time step in the set $\{0, \dots, T-1\}$ that describes the time step with respect to the cycle time T.

▶ **Algorithm 1.** The algorithm consists of three phases.

Phase 1: Initialization. For every vertex v and every time step $\theta \in \{0, \dots, T-1\}$, we create an array M_v^θ with $K+1$ entries. The entry $M_v^\theta[k]$ contains a label (t, k, v) with (absolute) arrival time t at v which has relative time θ and exactly k stops. We initialize every entry $M_v^\theta[k]$ with the label (∞, k, v) except for the entry $M_s^0[0]$ which is assigned the label $(0, 0, s)$. Furthermore, we create a priority queue $\mathcal{Q}$ and insert the label $(0, 0, s)$ into $\mathcal{Q}$. In $\mathcal{Q}$ the labels are ordered lexicographically.

Phase 2: Label Propagation. As long as $\mathcal{Q}$ is not empty, we extract the lexicographically smallest label (t, k, v) from $\mathcal{Q}$, i.e., for all other labels $(t', k', v') \in \mathcal{Q}$ it holds that either $t < t'$ or $t = t'$ and $k \leq k'$. Now we iterate through the outgoing edges of v. Let $e = (v, w)$ be such an edge.

1. If e has no traffic light, i.e., $e \notin E'$, then we create the label $(t + \tau(e), k, w)$.
2. If $e \in E'$ and $\texttt{green}(e, t) = 1$, then we also create the label $(t + \tau(e), k, w)$.
3. If $e \in E'$ and $\texttt{green}(e, t) = 0$, then we consider two cases:
   a. If we have an impatient cyclist: Let $\theta \in \{1, \ldots, T-1\}$ be the smallest number such that $\texttt{green}(e, t+\theta) = 1$; we create the label $(t + \theta + \tau(e), k+1, w)$.
   b. If we have a predictive cyclist: For *each* $\theta \in \{1, \ldots, T-1\}$ with $\texttt{green}(e, t+\theta) = 1$, we create the label $(t + \theta + \tau(e), k+1, w)$.

For each created label $L' = (t', k', v')$, we check whether it dominates another label. In particular, we check whether at vertex v', relative time $\theta' \equiv t' \bmod T$ and stop count k' the label at entry $M_{v'}^{\theta'}[k']$ has a worse (absolute) arrival time. If this is the case, then this label will be replaced by L' both in $M_{v'}^{\theta'}$ and in $\mathcal{Q}$.

Phase 3: Final Output. If we only want to solve Problem 2, then we can stop our algorithm as soon as the lexicographically smallest label in $\mathcal{Q}$ is at vertex d. We then return the absolute arrival time and the stop count of that label.

However, our algorithm is able to solve a more general problem. By running the algorithm until $\mathcal{Q}$ is empty we can give for each vertex v and each number of stops k the quickest walk (of impatient or predictive cyclists, respectively) from s to v with exactly k stops. This can be determined by running through every relative time step $\theta \in \{0, \ldots, T-1\}$ and search for the smallest absolute arrival time in the entries $M_v^\theta[k]$. We store these times in an array $\Omega \in \mathbb{N}^{|V| \times (K+1)}$.

Note that in Phase 2 we could also dominate labels where both the arrival time and the stop count is larger than in the constructed labels. However, we have refrained from doing so as we also want to compute the quickest routes with an exact number of stops given. Also note that although for a bounded number of stops the relaxed cyclists have no better arrival time than the predictive cyclists (see Lemma 5), for a fixed number of stops the relaxed cyclists might have a better arrival time as they can use a walk with less stops and better arrival time and stop somewhere unnecessarily. Similarly, the relaxed cyclists can achieve stop numbers that are infeasible for the predictive cyclists.

▶ **Theorem 10.** *Given a signalized network $N = (V, E, E', T, \tau, \gamma, \lambda)$, Algorithm 1 finds an array $\Omega \in \mathbb{N}^{|V| \times (K+1)}$ with the following property: If there is a timed s-v-walk in N with exactly k stops for the impatient or predictive cyclist, then $\Omega[v, k]$ contains the arrival time of the quickest of those walks for the respective cyclist.*

Proof. Firstly, let us shortly recall how a simple proof of Dijkstra's algorithm works. Let S be the set of vertices that were already processed by Dijkstra's algorithm, then one shows that the following two invariants hold during the algorithm:
1. for every vertex $v \in S$, the label of v is the length of a shortest path from s to v.
2. for every vertex $v \notin S$, the label of v is the length of a shortest path from s to v that only uses vertices of S as inner vertices.

As our algorithm processes a vertex several times, we have to adapt the proof. Instead of considering only vertices as elements of S, we consider tuples (θ, k, v) where $\theta \in \{0, \ldots, T-1\}$, $k \in \{0, \ldots, K\}$ and $v \in V$. We say that a timed walk $\mathcal{P} = (P, \pi)$ uses a tuple (θ, k, v) as *timed waypoint* if there is an edge $e_i = (w, v) \in P$ with $\pi(e_i) = t$ and $\theta \equiv (t + \tau(e_i)) \bmod T$ such that $\mathcal{P}$ has used exactly k stops when arriving at v at time step $t + \tau(e_i)$. Such a timed waypoint corresponds to the entry $M_v^\theta[k]$ in Algorithm 1.

Whenever the algorithm extracts a label (t, k, v) from $\mathcal{Q}$, we add the tuple $(t \bmod T, k, v)$ to S. We claim that the following two invariants hold during the algorithm.

1. For every tuple $(\theta, k, v) \in S$ let $M_v^\theta[k] = (t, k, v)$. Then t is the minimal arrival time of a timed s-v-walk in N that arrives at v at relative time step θ with exactly k stops.
2. For every tuple $(\theta, k, v) \notin S$ let $M_v^\theta[k] = (t, k, v)$. Then t is the minimal arrival time of a timed s-v-walk in N that arrives at v at relative time step θ with exactly k stops and only uses tuples of S as inner timed waypoint.

At the beginning, S is empty and both invariants trivially hold true. Now assume that both invariants hold and we add the new tuple $A = (\theta, k, v)$ to S. To show that the first invariant still holds true, it is sufficient to show that the invariant holds for the tuple A. Let t be the first entry of the label in $M_v^\theta[k]$. Assume there is a timed s-v-walk $\mathcal{P}$ that arrives at v before time step t. As the second invariant was true for A before A was added to S, $\mathcal{P}$ has to use some timed waypoint that is not in S. Let (θ', k', v') be the first timed waypoint on $\mathcal{P}$ that is not in S. There is a label (t', k', v') in $\mathcal{Q}$ with $t' \bmod T \equiv \theta'$. Due to the second invariant, t' is exactly the arrival time of path $\mathcal{P}$ at v'. The way the algorithm extracts the next label from $\mathcal{Q}$, $t' \geq t$ holds. This contradicts the choice of $\mathcal{P}$.

It remains the show that the second invariant holds after adding $A = (\theta, k, v)$ to S. If the minimal arrival time for walks using only elements of S as timed waypoints has decreased for some $B = (\theta', k', v') \notin S$ to the value t', then the new best timed s-v'-walk $\mathcal{P}'$ has to use the timed waypoint A as the second last waypoint. As was shown above, the arrival time in $M_v^\theta[k]$ is optimal. In Phase 2 of Algorithm 1 we have propagated this arrival time to the neighbors of v in an optimal manner, in particular to the vertex v'. Therefore, $M_{v'}^{\theta'}[k'] = t'$. ◀

Finally, we discuss the running time of the algorithm.

▶ **Proposition 11.** *For a given signalized network $N = (V, E, E', T, \tau, \gamma, \lambda)$, Algorithm 1 needs running time $\mathcal{O}(|V|^2 \cdot T^2 \cdot K)$ where K is the bound on the maximal number of stops.*

Proof. There are $\mathcal{O}(|V| \cdot T)$ arrays M_v^θ with $\mathcal{O}(K)$ entries each. Thus, these arrays have $\mathcal{O}(|V| \cdot T \cdot K)$ many entries in total. Every of those entries is extracted from the queue $\mathcal{Q}$ at most once. Using a heap, this extraction can be done in time $\mathcal{O}(\log(|V| \cdot T \cdot K)) \subseteq \mathcal{O}(|V| \cdot T)$ as K can be bounded by $|V|$, due to Proposition 7. After the extraction, we have to create at most $|V| \cdot T$ new labels[3] in Phase 2 of the algorithm and have to update their arrival times. This can be done in constant time per label, i.e., the total time for every extracted label is $\mathcal{O}(|V| \cdot T)$. This leads to the given running time bound. ◀

5 Conclusion

In this paper we have shown how we can use algorithmic ideas to avoid unwanted stops in bicycle routes. As we have seen, using cycles in routes and waiting at lights that are already green can be advantageous to avoid stops. Adding a constraint on the number of stops makes the route finding problem more difficult: The problem of finding the optimal route is then weakly NP-hard and we have presented a corresponding pseudo-polynomial algorithm. This new algorithm corrects a flaw in the approach of Chen and Yang [2].

[3] In real-world instances, this bound can be significantly improved since road networks usually have small vertex degrees.

Our results lead to a variety of new questions. In order to use our approach to design practically usable routings for cyclists one should take into account further aspects. Right now, the avoidance of stops may cause long detours. A better objective may be to look for a route that is at most, e.g., 10 % longer than the shortest path and has as few stops as possible. Instead of stopping, one may also consider slowing down or accelerating for a short period of time, that is, one could try to develop a model with variable speeds. Furthermore, we only have used fixed-time traffic signals so far, but also adaptive signals are commonly used in practice. However, this would add some uncertainty to the problem which can be addressed with the integration of techniques similar to those used in, e.g., [16] to find reliable shortest routes.

References

1 Ravindra K. Ahuja, James B. Orlin, Stefano Pallottino, and Maria Grazia Scutellà. Minimum time and minimum cost-path problems in street networks with periodic traffic lights. *Transportation Science*, 36(3):326–336, 2002. doi:10.1287/trsc.36.3.326.7827.

2 Yen-Liang Chen and Hsu-Hao Yang. Minimization of travel time and weighted number of stops in a traffic-light network. *European Journal of Operational Research*, 144(3):565–580, 2003. doi:10.1016/S0377-2217(02)00148-0.

3 Stuart E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17(3):395–412, 1969. doi:10.1287/opre.17.3.395.

4 Michael R. Garey and David S. Johnson. *Computers and Intractability*. W. H. Freeman, 29th edition, 2002.

5 Refael Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations research*, 17(1):36–42, 1992. doi:10.1287/moor.17.1.36.

6 P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. SCOOT – A traffic responsive method of coordinating signals. Technical Report Report No. LR 1014, Transport and Road Research Lab, Crowthorne, Berkshire, UK, 1981.

7 Alexander Kleff, Frank Schulz, Jakob Wagenblatt, and Tim Zeitz. Efficient Route Planning with Temporary Driving Bans, Road Closures, and Rated Parking Areas. In Simone Faro and Domenico Cantone, editors, *18th International Symposium on Experimental Algorithms (SEA 2020)*, volume 160 of *LIPIcs*, pages 17:1–17:13, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SEA.2020.17.

8 Ekkehard Köhler and Martin Strehler. Traffic signal optimization: Combining static and dynamic models. *Transportation Science*, 53(1):21–41, 2019. doi:10.1287/trsc.2017.0760.

9 Ekkehard Köhler, Rolf Möhring, Klaus Nökel, and Gregor Wünsch. Optimization of signalized traffic networks. In *Mathematics – Key Technology for the Future*, pages 179–188. Springer, 2008. doi:10.1007/978-3-540-77203-3_13.

10 Ariel Orda and Raphael Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM*, 37(3):607–625, 1990. doi:10.1145/79147.214078.

11 Ariel Orda and Raphael Rom. Minimum weight paths in time-dependent networks. *Networks*, 21(3):295–319, 1991. doi:10.1002/net.3230210304.

12 Dennis I. Robertson. TRANSYT: A traffic network study tool. Technical Report Report No. LR 253, Transport and Road Research Lab, Crowthorne, Berkshire, UK, 1969.

13 Markus Rogge. Berechnung kürzester Wege unter Berücksichtigung periodischer Ampelschaltungen für FahrradfahrerInnen. Master's thesis, Brandenburg University of Technology, Cottbus, Germany, 2021. In German.

14 Robert Scheffler and Martin Strehler. Optimizing traffic signal settings for public transport priority. In Gianlorenzo D'Angelo and Twan Dollevoet, editors, *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)*,

volume 59 of *OASIcs*, pages 9:1–9:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/OASIcs.ATMOS.2017.9.

15 Ipek N. Sener, Naveen Eluru, and Chandra R. Bhat. An analysis of bicycle route choice preferences in Texas, US. *Transportation*, 36:511–539, 2009. doi:10.1007/s11116-009-9201-4.

16 Liang Shen, Hu Shao, Ting Wu, Emily Zhu Fainman, and William H.K. Lam. Finding the reliable shortest path with correlated link travel times in signalized traffic networks under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 144:102159, 2020. doi:10.1016/j.tre.2020.102159.

17 Martin Strehler, Sören Merting, and Christian Schwan. Energy-efficient shortest routes for electric and hybrid vehicles. *Transportation Research Part B: Methodological*, 103:111–135, 2017. doi:10.1016/j.trb.2017.03.007.

18 Dean B. Taylor and Hani S. Mahmassani. Coordinating traffic signals for bicycle progression. *Transportation Research Record*, 1705(1):85–92, 2000. doi:10.3141/1705-13.

19 Tim Zeitz. NP-hardness of shortest path problems in networks with non-FIFO time-dependent travel times. *Information Processing Letters*, 179:106287, 2023. doi:10.1016/j.ipl.2022.106287.