Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 14, no. 2, pp. 123–147 (2010)

New complexity results for time-constrained dynamical optimal path problems

Sebastian Kluge[1] *Martin Brokate*[1] *Konrad Reif*[1,2]

[1]Technische Universität München, Lehrstuhl für mathematische Modellbildung, Boltzmannstraße 3, 85748 Garching b. München, Deutschland
[2]Baden-Wuerttemberg Cooperative State University, Ravensburg, Campus Friedrichshafen, Fallenbrunnen 2, 88045 Friedrichshafen, Deutschland

Abstract

In this paper, we consider time-dependent networks, and the task of computing cost-optimal paths, which are constrained to stay close to fastest paths. We derive pruning criteria, which significantly improve both the number of vertex-time pairs expanded during search and the memory required to ensure the correctness of any solution algorithm. We then prove new complexity results, which imply that the problem of computing constrained cost-optimal paths in a discrete-time setting is polynomially solvable for several graph and constraint classes.

Submitted:	Reviewed:	Revised:	Accepted:
June 2009	September 2009	October 2009	October 2009
	Final:	Published:	
	October 2009	January 2010	

Article type: Regular Paper
Communicated by: D. Wagner

E-mail addresses: kluge@ma.tum.de (Sebastian Kluge) brokate@ma.tum.de (Martin Brokate) reif@ma.tum.de (Konrad Reif)

1 Introduction

The problem of computing shortest paths in weighted and directed networks has been studied extensively in the literature. This is due to the fact, that this problem arises in many applications, such as route planning or internet routing, in the field of optimal control ([19]) and as subproblem in a variety of graph problems ([2]). The first algorithmical results were derived by Bellman [3] and Dijkstra [12], achieving a complexity of $\mathcal{O}(mn)$ and $\mathcal{O}(n^2 + m)$, respectively, were n denotes the number of vertices and m denotes the number of edges in the network. In the late 1960ies, a formal study of heuristic search algorithms ([18],[27]) began, and complexity results were derived, which depend on the length of the solution path ([26],[27],[25]). The problem of computing optimal paths in time-dependent networks was first introduced in [7], and indirectly mentioned in the context of maximal flows in [13]. This approach has received increasing attention since the 1990ies in the fields of intelligent transportation services [6] and internet routing [22]. As the computation of optimal paths in large networks, such as the road network, is still costly, a variety of speed-up techniques have been developed ([11]). There has also been considerable effort in extending the deterministic approach to a probabilistic setting, in which the edge travel times and costs are random variables or stochastic processes ([16], [17], [20], [15]).

The research in the field of time-dependent networks is divided into two approaches, a continuous (see, e.g., [23], [24], [9]) and a discrete (see, e.g., [4], [6], [1]) modelling of the time variable. The problems considered consist of computing optimal paths for one or for all travel times and from one or many source vertices to one or many goal vertices. Of course, depending on the model and the problem, different properties of the resulting dynamical network can be exploited, and different solution strategies have been developed.

Considering the complexity of the resulting solution algorithms, there are again two classes of problems: If the task consists of computing the fastest path for a fixed departure time in a network which fulfills the FIFO-condition (i.e., a network in which it is never possible to arrive earlier by leaving later), there exist algorithms which solve the problem in polynomial time ([23]). By contrast, if the FIFO-condition is violated, or minimum cost paths (with a cost different from travel time) are considered, the computation of optimal paths is NP-hard ([23], [1]). In the case of fastest paths, it is possible to overcome this hardness result, if unbounded waiting is allowed everywhere in the network ([23]). Yet, this approach is not applicable for minimum-cost paths.

In this work, we will consider a dynamical network, in which waiting is prohibited everywhere, and a dynamical optimal cost path shall be computed for a fixed departure time. This is a typical problem setting in applications like automotive navigation systems, in which the driver requests an optimal route to his desired destination, and repeated waiting in the road network is not allowed due to traffic constraints. We will discuss two constraints on dynamical paths, i.e., a time constraint and the claim that only simple paths are allowed for expansion. Note, that in contrast to static optimal paths, dynamical optimal paths may

contain circles [24]. We will show that both constraints can be verified in polynomial time. Moreover, we will derive pruning criteria, which allow a significant improvement in the number of vertex-time pairs, which must be expanded and which must be kept in memory by any solution algorithm. Finally, we derive new complexity results, which imply that constrained optimal cost paths can be computed in polynomial time, if the time variable is discrete. In case of a continuous time variable, we show that no time constraint other than allowing only fastest paths can ensure polynomial complexity in the worst case.
This paper is structured as follows: In Section 2, we introduce the notation required for the description of our problem, and give some preliminary results. In Section 3, we consider the computation of optimal paths in the absence of constraints, and show that the optimal cost function is Lipschitz-continuous, if the edge travel time and cost functions are. Based on this observation we derive a pruning criterion, which we extend to the time-constrained case in Section 4. Additionally, we show that in order to maintain the simple path property, not the whole history of a path, but only a small number of predecessors are required. In Section 5, we prove the complexity results for the computation of constrained dynamical optimal cost paths. Finally, we conclude our discussion of dynamical networks in Section 6.

2 Notation and problem formulation

Various ways of describing a dynamical network can be found in the literature [9], [4], [6], [24]. We find it convenient to use the following notation.

Definition 1 *A dynamical network is a quadruple $(V, E, \tau; \beta)$, where V is a set of n vertices ($n \in \mathbb{N}$), $E \subset V \times V$ is a set of m directed edges ($m \in \mathbb{N}$), $\tau : E \times \mathbb{R} \to \mathbb{R}_0^+$ is an edge travel time function and $\beta : E \times \mathbb{R} \to \mathbb{R}$ is an edge cost function.*
Given a directed edge $e = (u, v) \in E$, we denote by $\alpha : E \to V$, $\alpha(e) = u$ the tail of the edge e and by $\omega : E \to V$, $\omega(e) = v$ the head of the edge e.

Remark 1 *The second argument of τ and β, respectively, denotes the time variable and refers to the departure time on the edge given by the first argument.*

A state in the dynamical network must therefore be specified by a vertex and a time, whereas a state transition is specified by an edge and the corresponding travel time. This notion leads to the following definition of dynamical paths.

Definition 2 *A dynamical path p of length $l \in \mathbb{N}$ is a sequence of pairs $(e_k, s_k) \in E \times \mathbb{R}$, $k = 1, ..., l$, i.e., $p = ((e_1, s_1), (e_2, s_2), ..., (e_l, s_l))$, with the following properties:*

$$\alpha(e_{k+1}) = \omega(e_k), \quad k = 1, ..., l-1, \tag{1}$$

$$s_{k+1} = s_k + \tau(e_k, s_k), \quad k = 1, ..., l-1. \tag{2}$$

s_1 is called the departure time of the path p, $\alpha(e_1)$ is called the source vertex, $\omega(e_l)$ is called the goal vertex.
We denote the set of all dynamical paths of finite length l by $\mathcal{P}$ and the set of finite connected edge sequences (i.e., edge sequences satisfying (1)) by $\mathcal{E}$.

Remark 2 *The set $\mathcal{E}$ is the set of all finite topological paths in (V, E).*

Given a connected sequence of edges $(e_1, ..., e_l) \in \mathcal{E}$ and a starting time $s_1 \in \mathbb{R}$, equation (2) uniquely determines a dynamical path. This motivates the definition of a path projection $\Pi : \mathcal{P} \to \mathbb{R} \times \mathcal{E}$, with components $\Pi_1 : \mathcal{P} \to \mathbb{R}$, $\Pi_2 : \mathcal{P} \to \mathcal{E}$, $\Pi = (\Pi_1, \Pi_2)$ and

$$\Pi_1\left((e_1, s_1), (e_2, s_2), ..., (e_l, s_l)\right) = s_1, \tag{3}$$

$$\Pi_2\left((e_1, s_1), (e_2, s_2), ..., (e_l, s_l)\right) = (e_1, e_2, ..., e_l). \tag{4}$$

The path projection Π and its inverse will considerably facilitate the notation and the treatment of dynamical paths corresponding to the same edge sequence at varying departure times in Section 3.
Next, we define the path travel time function $t : \mathcal{P} \to \mathbb{R}$,

$$t(p) = \sum_{k=1}^{l} \tau(e_k, s_k). \tag{5}$$

Definition 3 *Let $\mathcal{P}(v; u, s)$ denote the set of dynamical paths from vertex $u \in V$ to vertex $v \in V$ with departure time $s \in \mathbb{R}$.*
We define the optimal travel time function $t^* : V \times (V \times \mathbb{R}) \to \mathbb{R}_0^+$,

$$t^*(v; u, s) = \begin{cases} \inf\{t(p) : p \in \mathcal{P}(v; u, s)\}, & u \neq v \\ 0, & u = v \end{cases}. \tag{6}$$

Each path $p^* \in \mathcal{P}(v; u, s)$ with $t(p^*) = t^*(v; u, s)$ is called a fastest path from u to v with respect to the departure time s.

In static networks with nonnegative edge cost, optimal paths are always simple. They can be computed, e.g., applying the principle of dynamic programming or the algorithm of Dijkstra. In a time-dependent network, the principle of dynamic programming is only generally valid in the time-expanded network ([1], [10]). This explains the difficulty of deriving computationally efficient algorithms for the dynamical optimal path problem: The time-expanded network is usually very large in the case of a discrete time variable, and even the set of reachable vertex-time pairs is eventually innumerable in the case of a continuous time variable ([24]). Several pseudo-polynomial algorithms have been developed for discrete-time time-expanded networks, exploiting the fact that the time-expanded network is acyclic if all travel times are positive ([1], [6], [4]). For some applications, like automotive navigation systems, it might be desirable to exclude circles in the topological structure of paths. This is on one hand motivated by the smaller number of feasible paths, which must be considered during search, and which will in almost all cases suffice for the computation of optimal paths. On the other hand, it is unlikely that an optimal path which contains a circle will be accepted by the driver. We will now define which dynamical paths we call simple.

Definition 4 *A dynamical path $p \in \mathcal{P}$ is called simple, if the topological path $\Pi_2(p)$ is simple, i.e., if $\Pi_2(p)$ visits no vertex more than once.*

In this work, we will consider a dynamical network, in which all edge travel times fulfill the FIFO-condition, i.e., we suppose that for all $s, s' \in \mathbb{R}$, $s' \geq s$, and all $e \in E$, we have

$$s' + \tau(e, s') \geq s + \tau(e, s). \tag{7}$$

The FIFO-property states, that it is not possible to arrive earlier by leaving later. In traffic theory this is also referred to as the non-passing property ([28]). The FIFO-property has an important impact on the structure of fastest paths and on the complexity of computing the like.

Lemma 1 *Suppose that the edge travel times of the dynamical network fulfill the FIFO-condition (7). Then for every source vertex $v_0 \in V$, every departure time $s_0 \in \mathbb{R}$ and every goal vertex $v' \in V$, there exists a simple and concatenated fastest path, and the computation of this fastest path can be carried out in $\mathcal{O}(m + n\log(n))$ time.*

Proof: The existence of a simple and concatenated fastest path has been proved in [23, Corollary 1], whereas the relevance of the FIFO-property is explicitly stated in [23, Section 3.2]. A slightly modified version of Dijkstra's shortest path algorithm ([12]) can be used to compute the fastest path in a FIFO network ([1]). Using Fibonacci heap implementation ([14]), this algorithm can be implemented in $\mathcal{O}(m + n\log(n))$ time. □

Remark 3 *Note, that in sparse networks, i.e., networks in which $m = \mathcal{O}(n)$, the complexity bound of Lemma 1 becomes $\mathcal{O}(n\log(n))$. The road network, in which the number of roads emanating from any junction is bounded, is a sparse network.*

Fastest paths in dynamical networks are therefore simple and easy to compute. This motivates the introduction of the second constraint, which requires any feasible path to remain in some sense close to a fastest path.

Definition 5 *Let $\gamma : \mathbb{R} \to \mathbb{R}$ denote a monotonically increasing function with $\gamma(0) = 0$, and let $\Gamma : \mathbb{R} \to \mathbb{R}$, $\Gamma(s) = s + \gamma(s)$. Given a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$, we call the visiting time $s \in \mathbb{R}$ feasible for the vertex $v \in V$, if*

$$t^*(v; v_0, s_0) \leq s, \tag{8}$$
$$s \leq \Gamma(t^*(v; v_0, s_0)), \tag{9}$$
$$s + t^*(v'; v, s) \leq \Gamma(t^*(v'; v_0, s_0)). \tag{10}$$

Remark 4 *The constraint (9) states, that it is not feasible to visit a vertex more than $\gamma(t^*(v; v_0, s_0))$ after the optimal travel time $t^*(v; v_0, s_0)$. Considering automotive navigation systems, it may, e.g., be prohibitive to compute a route, which takes more than 110% of the optimal travel time.*
The inequalities (8) and (10) guarantee, that it is possible to reach v at (or before) time s, and that the goal vertex is reachable at a feasible visiting time from v at (or after) time s.

Remark 5 *Since the set of feasible visiting times for any vertex $v \in V$ depends on the source vertex v_0, the departure time s_0 and the goal vertex v' (cf. Definition 5), these must be specified whenever we deal with time-constrained dynamical optimal path problems. Even if not explicitly noted, we will assume that some particular v_0, s_0, s' have been fixed whenever time constraints are imposed.*

In the following, we will consider two classes of time constraints, i.e., we will choose γ as a linear function or as a logarithmic function. For the sake of simplicity, we denote

$$\gamma_{\text{lin}}(s) = s, \tag{11}$$
$$\gamma_{\log}(s) = \log(s), \tag{12}$$

where log denotes the natural logarithm. Note, that the introduction of constants or the choice of a logarithmic function to a different basis will not result in different orders of complexity in Section 5. Hence, the functions $\gamma_{\text{lin}}, \gamma_{\log}$ can be viewed as representants for a whole class of functions. These classes have been chosen in analogy to the literature, which investigates the effect of the accuracy of a given heuristic on the complexity of heuristic search (cf. Section 5). Yet, this choice is somewhat arbitrary, and results similar to those of Lemma 4, Theorem 3 and Corollary 3 can also be achieved for other function classes.
In an unconstrained optimal path problem, none of the two constraints (i.e., the simple path constraint and the feasible visiting time constraint) is imposed. As each constraint has a different impact on the complexity of computing optimal paths, we will seperately discuss the effects of the constraints in both continuous and discrete time. Generally, depending on the set of constraints we impose on dynamical paths, we will result in a set of feasible dynamical paths, which we denote by $\hat{\mathcal{P}}$. Note, that in the unconstrained case, there holds $\hat{\mathcal{P}} = \mathcal{P}$.
In the same manner, in which we have defined a path travel time function, we now define a path cost function $b : \hat{\mathcal{P}} \to \mathbb{R}$ for each feasible path $p = ((e_1, s_1), (e_2, s_2), ..., (e_l, s_l)) \in \hat{\mathcal{P}}$,

$$b(p) = \sum_{k=1}^{l} \beta(e_k, s_k). \tag{13}$$

Definition 6 *Let $\hat{\mathcal{P}}(v; u, s) \subset \mathcal{P}$ denote the set of feasible dynamical paths from vertex $u \in V$ to vertex $v \in V$ with departure time s. Let*

$$S(v; u) = \begin{cases} \{s \in \mathbb{R} : \hat{\mathcal{P}}(v; u, s) \neq \emptyset\}, & u \neq v \\ \{s \in \mathbb{R} : s \textit{ satisfies all imposed time constraints at } u = v\}, & u = v \end{cases},$$

and $\mathbb{D} = \{(v; u, s) \in V \times (V \times \mathbb{R}) : s \in S(v; u)\}$.
We define the optimal cost function $b^* : \mathbb{D} \to \mathbb{R}$,

$$b^*(v; u, s) = \begin{cases} \inf\{b(p) : p \in \hat{\mathcal{P}}(v; u, s)\}, & u \neq v \\ 0, & u = v \end{cases} . \tag{14}$$

Each path $p^* \in \hat{\mathcal{P}}(v; u, s)$ *with* $b(p^*) = b^*(v; u, s)$ *is called an optimal path from* u *to* v *with respect to the departure time* s.

Remark 6 *For* $u \neq v$, $S(v; u)$ *denotes the set of visiting times, for which there exists at least one feasible dynamical path from* u *to* v. *Note, that in the absence of time constraints, there holds* $S(v; u) = \mathbb{R}$ *for all* $u, v \in V$, *and hence* $b^* : V \times (V \times \mathbb{R}) \to \mathbb{R}$.

We are now ready to formulate the problem of computing the minimum cost feasible dynamical path.

Problem 1 *Given a source vertex* $v_0 \in V$, *a departure time* $s_0 \in \mathbb{R}$ *and a goal vertex* $v' \in V$, *determine a feasible path* $p^* \in \hat{\mathcal{P}}(v'; v_0, s_0)$ *with* $b(p^*) = b^*(v'; v_0, s_0)$.

3 The unconstrained problem

In this section, we will consider the problem of computing unconstrained optimal paths. This problem is - at least from an algorithmic point of view - somewhat simpler than the computation of constrained optimal paths, but some of the results we will obtain are the basis for further considerations in the time-constrained case.
We have restricted ourselves to finite dynamical paths in Definition 2. However, if no additional assumptions are imposed on the dynamical network, a finite optimal path does not have to exist. If, for example, the topological network is not strongly connected, then there is at least one pair of vertices $v_0, v' \in V$, such that there exists no path from v_0 to v'. Even if the network is strongly connected, there might be no finite optimal path, but rather an infinite optimal policy, which describes how to reach the goal vertex with minimum cost ([24]). We therefore need some additional assumptions, which ensure the existence of optimal paths.

Assumption 1 *The topological network* (V, E) *is strongly connected and finite. The edge costs are bounded by positive constants* $\underline{\beta}, \overline{\beta} \in \mathbb{R}$,

$$\underline{\beta} \leq \beta(e, s) \leq \overline{\beta}, \quad \forall e \in E, \, s \in \mathbb{R}. \tag{15}$$

Remark 7 *Note, that the formulation of the upper bound in (15) is not necessary if both edge travel time and edge cost functions are continuous, or there exists at least one path of finite cost for any choice of* v_0, s_0 *and* v'. *Yet, some of the following analysis would become more involved, and hence (15) has been assumed for simplicity.*

Theorem 1 *If Assumption 1 holds, then there exists at least one optimal path from any source vertex $v_0 \in V$ to any goal vertex $v' \in V$ and for any departure time $s_0 \in \mathbb{R}$. Moreover, the set*

$$\mathcal{E}_{opt}(v'; v_0) = \{\Pi_2(p) : b(p) = b^*(v'; v_0, s), p \in \mathcal{P}(v'; v_0, s), s \in \mathbb{R}\} \tag{16}$$

is finite for all $v_0, v' \in V$.

Proof: Let $v_0, v' \in V$, $s_0 \in \mathbb{R}$ be arbitrary but fixed. As (V, E) is strongly connected and finite, the minimum-hop distance d from v_0 to v' is finite, $d \in \mathbb{N}$. Let $(e_1, ..., e_d)$ be a topological minimum-hop path from v_0 to v'. From (13) and (15) we deduce

$$d\overline{\beta} \geq b\left(\Pi^{-1}(s_0, (e_1, ..., e_d))\right) \geq \inf\{b(p) : p \in \mathcal{P}(v'; v_0, s_0)\} = b^*(v'; v_0, s_0), \tag{17}$$

hence the cost of the optimal path from v_0 to v' with departure time s_0 is bounded. The cost of any optimal path is therefore bounded from above by $n\overline{\beta}$, and hence the length of any optimal path is bounded by $n\overline{\beta}/\underline{\beta}$. As the set of dynamical paths of length $l \leq n\overline{\beta}/\underline{\beta}$ emanating from v_0 at time s_0 is finite, at least one element of this set must be optimal. Noting that the bound $l \leq n\overline{\beta}/\underline{\beta}$ is independent of the departure time, this also implies that the set of edge sequences $\mathcal{E}_{\text{opt}}(v'; v_0)$ is finite. □

Remark 8 *The set $\mathcal{E}_{opt}(v'; v_0)$ contains all topological paths ϵ from u to v, which define an optimal dynamical path $p = \Pi^{-1}(s, \epsilon)$ for some departure time $s \in \mathbb{R}$. We will need this set to prove the continuity of the optimal cost function in Lemma 2.*

We will now show, that under stronger structural assumptions, i.e., Lipschitz continuitiy of the edge travel time and edge cost functions, the optimal cost function defined in (14) is Lipschitz-continuous. The following lemma is fundamental for the derivation of the path pruning principle, which we will subsequently present, and which we will extend to time-constrained optimal paths in Section 4.

Lemma 2 *Consider a dynamical network in which Assumption 1 holds. If τ, β are Lipschitz-continuous in the second argument with constants $L_\tau, L_\beta > 0$, i.e.,*

$$|\tau(e, s) - \tau(e, s')| \leq L_\tau |s - s'|, \quad \forall e \in E, \, s, s' \in \mathbb{R}, \tag{18}$$

$$|\beta(e, s) - \beta(e, s')| \leq L_\beta |s - s'|, \quad \forall e \in E, \, s, s' \in \mathbb{R}, \tag{19}$$

then for any $u, v \in V$ the partial function $b^*(v; u, .) : \mathbb{R} \to \mathbb{R}$ is Lipschitz-continuous with Lipschitz-constant

$$L = L_\beta \frac{(1 + L_\tau)^D - 1}{L_\tau}, \tag{20}$$

where D denotes the maximum length of an optimal path from from u to v, i.e.,

$$|b^*(v; u, s) - b^*(v; u, s')| \leq L|s - s'|, \quad \forall s, s' \in \mathbb{R}. \tag{21}$$

Proof: Let $p \in \mathcal{P}(v;u,s)$ be a dynamical path from $u \in V$ to $v \in V$ with departure time $s \in \mathbb{R}$, $p = ((e_1,s_1),...,(e_l,s_l))$, $s_1 = s$, $l \leq D$. For $s' \in \mathbb{R}$ let p' denote the dynamical path corresponding to the same edge sequence $\epsilon = \Pi_2(p)$ and the departure time s', i.e., $p' = \Pi^{-1}(s',\epsilon) = ((e_1,s'_1),...,(e_l,s'_l))$, $s'_1 = s'$. From (2) and (18), we have for $k = 2,...,l$, that

$$\begin{aligned} |s_k - s'_k| &= |s_{k-1} + \tau(e_{k-1},s_{k-1}) - s'_{k-1} - \tau(e_{k-1},s'_{k-1})| \\ &\leq (1+L_\tau)|s_{k-1} - s'_{k-1}|. \end{aligned}$$

Inductively, it follows that

$$|s_k - s'_k| \leq (1+L_\tau)^{k-1}|s - s'|, \quad k = 1,...,l. \tag{22}$$

Using (19) and (22), we derive

$$\begin{aligned} |b(p) - b(p')| &= \left|\sum_{k=1}^{l} \beta(e_k,s_k) - \sum_{k=1}^{l} \beta(e_k,s'_k)\right| \leq \sum_{k=1}^{l} L_\beta |s_k - s'_k| \\ &\leq \sum_{k=1}^{l} L_\beta (1+L_\tau)^{k-1}|s-s'| \leq \sum_{k=1}^{D} L_\beta (1+L_\tau)^{k-1}|s-s'| \\ &= L_\beta \frac{(1+L_\tau)^D - 1}{L_\tau}|s-s'|, \end{aligned}$$

where the last equality follows from the formula for the geometric series. Hence the mapping $s \mapsto b(\Pi^{-1}(s,\epsilon))$ is Lipschitz-continuous with Lipschitz constant L for every topological path ϵ from u to v, and arbitrary departure time $s \in \mathbb{R}$. According to Theorem 1, $\mathcal{E}_{\text{opt}}(v;u)$ contains only a finite number of elements. By construction, for any $s \in \mathbb{R}$ there exists a $\epsilon^* \in \mathcal{E}_{\text{opt}}(v;u)$, such that $b^*(v;u,s) = b(\Pi^{-1}(s,\epsilon^*))$. We can therefore write the optimal cost function defined in (14) as

$$\begin{aligned} b^*(v;u,s) &= \inf\{b(\Pi^{-1}(s,\Pi_2(p))) : p \in \mathcal{P}(v;u,s)\} \\ &= \min\{b(\Pi^{-1}(s,\epsilon)) : \epsilon \in \mathcal{E}_{\text{opt}}(v;u)\}. \end{aligned}$$

Hence the partial function $b^*(v;u,\cdot)$ is the pointwise minimum of a finite number of Lipschitz-continuous functions, and thus Lipschitz-continuous with Lipschitz-constant L. □

Remark 9 *From the proof of Theorem 1, it follows, that the length of any optimal path is bounded from above by $n\overline{\beta}/\underline{\beta}$. In practical applications, there are usually a plurality of more sophisticated techniques for the derivation of an upper bound for the length of each optimal path from u to v, such as, e.g., using landmarks ([11]): Suppose, that upper bounds $\overline{b}$ for the optimal cost functions b^* with respect to a landmark v^* are given for two vertices $u,v \in V$, i.e., we know that $b^*(v^*;u,s) \leq \overline{b}(v^*;u)$ and $b^*(v;v^*,s) \leq \overline{b}(v;v^*)$ for all $s \in \mathbb{R}$. Then, as a consequence of the triangle inequality, we also obtain $b^*(v;u,s) \leq \overline{b}(v;v^*) + \overline{b}(v^*;u)$. Note, that a smaller Lipschitz constant will result in a stronger pruning criterion.*

As we have pointed out in Section 2, the computation of a solution to the dynamical optimal path problem must be carried out in the time-expanded network. Especially when the edge travel times are functions of a continuous time variable, most paths from v_0 to a vertex $v \in V$ with departure time s_0 will result in different arrival times. As the time-expanded network may contain a large number of vertex-time-pairs in a small time interval, it is of high practical interest to prune any vertex-time-pair, which cannot be contained in an optimal path. Although this is particularly important in the case of a continuous time variable, in which a large number of vertex-time-pairs may be contained in an arbitrarily small time interval, the following result holds also in the case of a discrete time variable.

Lemma 3 *Consider a dynamical network in which Assumption 1 holds, and let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given. Suppose, that τ, β are Lipschitz-continuous in the second argument with constants $L_\tau, L_\beta > 0$.*
For a given vertex $v \in V$, let $D = d\overline{\beta}/\underline{\beta}$, where d denotes the minimum-hop distance from v to v', and L according to (20). If $p, p' \in \mathcal{P}(v; v_0, s_0)$, then p' cannot be extended to an optimal path, if

$$b(p') > b(p) + L|t(p) - t(p')|. \tag{23}$$

Proof: As in the proof of Theorem 1 (cf. equation (17)), we see that the length of any optimal path from any vertex-time pair (v, s) to v' is bounded from above by $D = d\overline{\beta}/\underline{\beta}$. Applying Lemma 2, the partial function $b^*(v'; v, \cdot) : \mathbb{R} \to \mathbb{R}$ is Lipschitz-continuous with the Lipschitz-constant L given by (20). The minimum-cost extension of a path $p \in \mathcal{P}(v; v_0, s_0)$ which leads to the goal vertex v' is the extension by an optimal path from v to v' with departure time $s = s_0 + t(p)$. Consequently, using the Lipschitz-continuity of $b^*(v'; v, \cdot)$, (23) implies that

$$\begin{aligned} b(p) + b^*(v'; v, s_0 + t(p)) &\leq b(p) + b^*(v'; v, s_0 + t(p')) + L|t(p) - t(p')| \\ &< b(p') + b^*(v'; v, s_0 + t(p')). \end{aligned}$$

Therefore, p' cannot be extended to an optimal path. □

The following simple example illustrates the use of the path pruning criterion: Consider the dynamical network given by the graph in Figure 1, with

$$\tau(e_0, s) = 0.1\,,$$
$$\beta(e_0, s) = 0.5\,.$$

Suppose, that τ, β are Lipschitz-continuous in the second argument with constants $L_\tau = L_\beta = 0.15$, and $\beta(e, s) \geq \underline{\beta} = 0.5$. Let $s_0 = 0$, and consider the task of computing the cost-optimal dynamical path from v_0 to v' with departure time s_0. We assume, that (e.g., from a static preprocessing step) we know that $b^*(v'; v_0, s) \leq 5$ for all $s \in \mathbb{R}$. This implies, that the topological

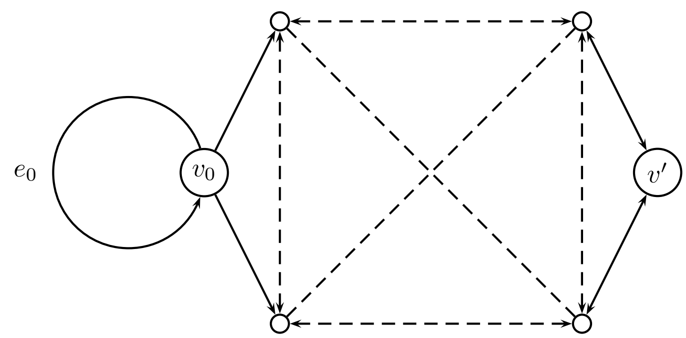

Figure 1: Topological structure of the example network. The dashed center part of the graph may be arbitrary, but such that there exists at least one topological path from v_0 to v'. It might be, e.g., a symmetric grid graph of arbitrary size.

length of an optimal path is bounded from above by $D = b^*(v'; v_0, s_0)/\underline{\beta} \leq 10$. Consequently, the partial mapping $s \mapsto b^*(v'; v_0, s)$ is Lipschitz-continuous with Lipschitz-constant

$$L = L_\beta \frac{(1 + L_\tau)^D - 1}{L_\tau} \leq 3.1\,.$$

As the optimal path may contain circles, we must generally consider all copies of the source vertice v_0 in the time-expanded network. Since $b^*(v'; v_0, s) \leq 5$ we must eventually consider 11 copies of v_0 if the vertex-time pairs are expanded in an increasing order of cost. Let $p_k = ((e_0, 0), ..., (e_0, (k-1) \cdot 0.1))$ denote the dynamical path k times cycling e_0. In addition to $(v_0, 0)$ (which may be considered as reached by the path p_0 of length 0 emanating from v_0), the vertex-time pairs $(v_0, k \cdot 0.1)$ are reached by p_k, $k = 1, ..., 10$, respectively. The travel times and costs associated with p_k, $k = 0, ..., 10$, are

$$t(p_k) = k \cdot 0.1\,,$$
$$b(p_k) = k \cdot 0.5\,.$$

Now, since

$$b(p_k) = k \cdot 0.5 > L \cdot k \cdot 0.1 = b(p_0) + L|t(p_k) - t(p_0)|,$$

Lemma 3 implies that p_k cannot be extended to an optimal path, if $k = 1, ..., 10$. Hence, only by considering the source vertex, the application of the path pruning criterion has significantly reduced the size of the search space. Instead of 11 possible copies of v_0 in the time-expanded network, only $(v_0, 0)$ needs to be considered for the computation of the optimal dynamical path. Of course, the same procedure can be repeated in any subsequent vertex, resulting in a further reduction of the search space. Although this is only an illustrative example, and the performance of the pruning criterion depends on the underlying network and the particular application, it shows the potential of the simple test given by equation (23).

4 Pruning principles for the constrained optimal path problems

In this Section we will derive two criteria, which define admissible pruning strategies for the computation of constrained optimal paths. We will first extend the result of Lemma 3 to the time-constrained case and then derive a pruning criterion which significantly decreases the complexity of maintaining the simple path property.
Let us give a precise definition of feasible time-constrained paths:

Definition 7 *Given a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$, we call a dynamical path $p \in \mathcal{P}$ feasible, if it visits vertices only at feasible times, i.e., if we have for $p = ((e_1, s_1), (e_2, s_2), ..., (e_l, s_l))$, that*

$$s_k \in S(v'; \alpha(e_k)), \quad k = 1, ..., l, \tag{24}$$

$$s_1 + t(p) \in S(v'; \omega(e_l)). \tag{25}$$

Remark 10 *As the fastest path from v_0 to v' with departure time s_0 is simple (cf. Lemma 1) and passes vertices only at feasible visiting times (cf. Definition 5), there exists at least one feasible path from v_0 to v' with departure time s_0: This holds also, if in addition to the time constraint, paths are constrained to be simple.*

In order to decide, whether a dynamical path is feasible or not, it is necessary to know the set of feasible visiting times $S(v'; v)$ for all $v \in V$. This is based on the knowledge of a large number of optimal travel times (cf. Definition 5). We will therefore state the following assumption, which not only implies the finiteness of fastest and optimal paths, but is also the basis for the complexity results in Section 5.

Assumption 2 *The topological network (V, E) is strongly connected and finite. The edge travel times fulfill the FIFO-condition and are bounded by positive constants $\underline{\tau}, \overline{\tau} \in \mathbb{R}$,*

$$\underline{\tau} \leq \tau(e, s) \leq \overline{\tau}, \quad \forall e \in E,\ s \in \mathbb{R}. \tag{26}$$

Theorem 2 *If Assumption 2 holds, then there exists at least one optimal path from any source vertex $v_0 \in V$ to any goal vertex $v' \in V$ and for any departure time $s_0 \in \mathbb{R}$. Moreover, the set $\mathcal{E}_{feas}(v; u) = \{\Pi_2(p) : p \in \hat{\mathcal{P}}(v; u, s), s \in S(v; u)\}$ is finite for all $u, v \in V$.*

Proof: Let $v_0, v' \in V$, $s_0 \in \mathbb{R}$ be arbitrary but fixed. As (V, E) is strongly connected and finite, the minimum-hop distance d from v_0 to v' is finite, $d \in \mathbb{N}$. Let $(e_1, ..., e_d)$ be a topological minimum-hop path from v_0 to v'. From (5) and (26) we deduce

$$d\overline{\tau} \geq t\left(\Pi^{-1}(s_0, (e_1, ..., e_d))\right) \geq \inf\{t(p) : p \in \hat{\mathcal{P}}(v'; v_0, s_0)\} = t^*(v'; v_0, s_0),$$

hence the travel time of the fastest path from v_0 to v' with departure time s_0 is bounded. The maximum feasible visiting time for any vertex in the network is therefore bounded from above by $\Gamma(d\overline{\tau})$, and hence the length of any feasible path is bounded by $\Gamma(d\overline{\tau})/\underline{\tau}$. The remaining part of the proof follows as in the proof of Theorem 1. □

Remark 11 *The set $\mathcal{E}_{feas}(v;u)$ contains all topological paths ϵ from u to v, which define a feasible dynamical path $p = \Pi^{-1}(s,\epsilon)$ for some departure time $s \in \mathbb{R}$. With respect to continuity, it plays a similar role to $\mathcal{E}_{opt}(v;u)$ in the unconstrained case.*

Remark 12 *The result of Theorem 2 holds even in a dynamical network, in which the FIFO-property is not fulfilled. Yet, the assumption of the FIFO-property will be crucial for the derivation of the complexity results in Section 5. Note also, that the edge cost function may assume arbitrary values.*

The optimal cost function is not necessarily continuous in the case of time-constrained optimal paths. This is due to the fact, that an edge sequence $\epsilon \in \mathcal{E}_{\text{feas}}(v;u)$ may produce very low values of the cost function but become infeasible at a certain time σ, due to the constraint on the visiting times. In such a case the optimal cost function would jump to the value defined by the next-best feasible path. (Note, that due to the FIFO-property the number of feasible edge sequences can only decrease as time increases.) We therefore only have $\lim_{s\uparrow\sigma} b^*(v;u,s) \leq \lim_{s\downarrow\sigma} b^*(v;u,s)$, with a finite number of jumps (because $\mathcal{E}_{\text{feas}}(v;u)$ is finite, cf. Theorem 2). As in the case of continuous edge travel times the feasible time intervals $S(v;u)$ are closed for all $u, v \in V$, the optimal cost function is continuous from the left, provided that all edge cost functions are continuous (see Figure 4). This leads to the following extension of Lemma 3.

Corollary 1 *Consider a dynamical network in which Assumption 2 holds, and let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given. Suppose, that τ, β are Lipschitz-continuous in the second argument with constants $L_\tau, L_\beta > 0$.*
For a given vertex $v \in V$, let $D = [\Gamma(t^*(v';v_0,s_0)) - t^*(v;v_0,s_0)]/\underline{\tau}$, and L according to (20). If $p, p' \in \hat{\mathcal{P}}(v;v_0,s_0)$, then p' cannot be extended to an optimal path, if $t(p') \geq t(p)$ and

$$b(p') > b(p) + L(t(p') - t(p)). \tag{27}$$

Proof: As a feasible path $p \in \hat{\mathcal{P}}$ from v to v' must depart and arrive at feasible times, its travel time is bounded by $\max S(v';v') - \min S(v';v) = \Gamma(t^*(v';v_0,s_0)) - t^*(v;v_0,s_0)$. The length of such a feasible path is therefore bounded by $D = [\Gamma(t^*(v';v_0,s_0)) - t^*(v;v_0,s_0)]/\underline{\tau}$. Applying Lemma 2, the partial function $b^*(v';v,.) : S(v';v) \to \mathbb{R}$ is Lipschitz-continuous with the Lipschitz-constant L given by (20) on every time interval $S \subset S(v';v)$, which contains no discontinuity. Note, that $S(v';v)$ is also an interval, as the edge travel times are

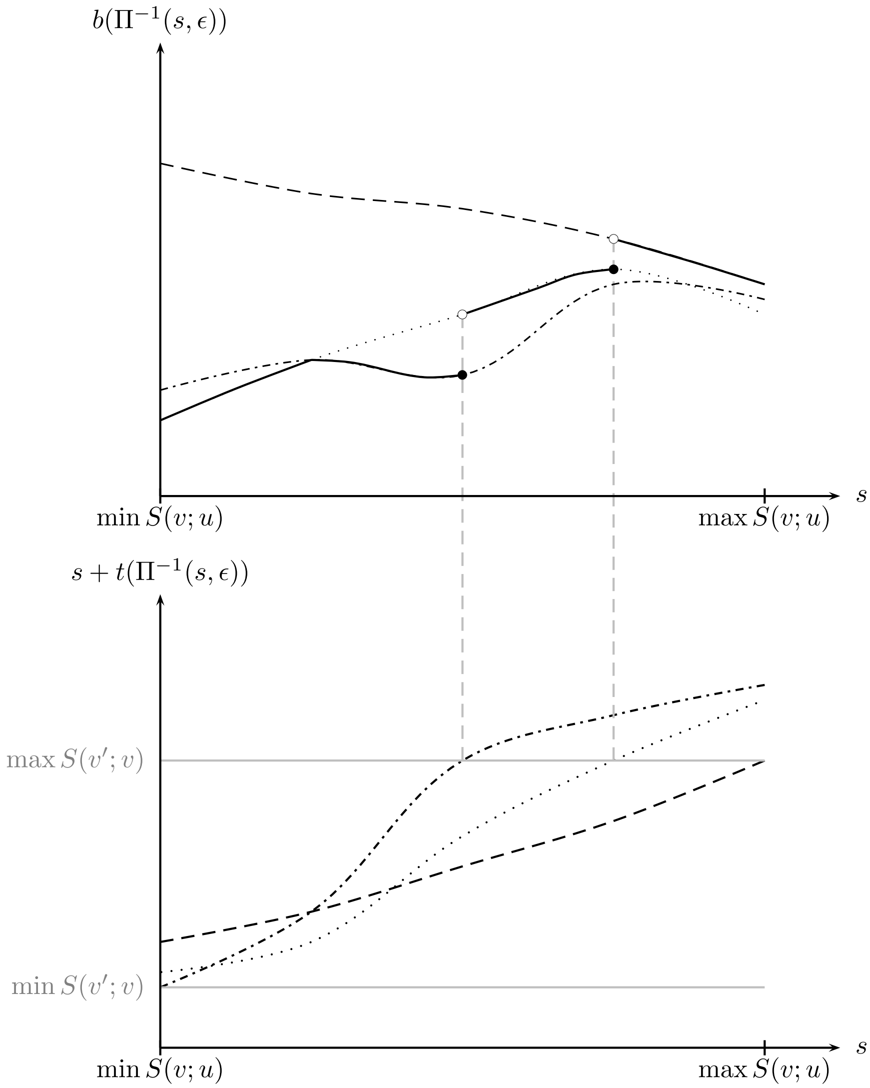

Figure 2: Cost functions and arrival time functions of dynamical paths, corresponding to three topological paths from vertex u to vertex v and varying departure times s (dashed, chain-dotted, dotted black curves). The grey line in the lower drawing constitutes the time constraint in v, the solid black curve in the upper drawing illustrates the resulting constrained optimal cost function.

continuous. Let $\sigma_1, ..., \sigma_j$, $j \in \mathbb{N}$, denote the time instants, at which $b^*(v'; v, .)$ is discontinuous, and let $\beta_i = \lim_{s\downarrow\sigma_i} b^*(v'; v, s) - \lim_{s\uparrow\sigma_i} b^*(v'; v, s)$, $i = 1, ..., j$, denote the height of the i-th jump. As we have argued before, $\beta_i > 0$ for all $i = 1, ..., j$. Consequently, for $t(p') \geq t(p)$, there holds

$$\begin{aligned} b^*(v'; v, s_0 + t(p')) &\geq b^*(v'; v, s_0 + t(p)) - L(t(p') - t(p)) + \sum_{i:s\leq\sigma_i<s'} \beta_i \\ &\geq b^*(v'; v, s_0 + t(p)) - L(t(p') - t(p)). \end{aligned} \tag{28}$$

The minimum-cost extension of a path $p \in \hat{\mathcal{P}}(v; v_0, s_0)$ which leads to the goal vertex v' is the extension by an optimal path from v to v' with departure time $s = s_0 + t(p)$. Consequently, (27) and (28) imply that

$$\begin{aligned} b(p) + b^*(v'; v, s_0 + t(p)) &\leq b(p) + b^*(v'; v, s_0 + t(p')) + L(t(p') - t(p)) \\ &< b(p') + b^*(v'; v, s_0 + t(p')). \end{aligned}$$

Therefore, p' cannot be extended to an optimal path. □

Let us now consider the case, in which in addition to the time constraint, feasible paths are constrained to be simple. In this case, any solution algorithm will have to remember the history of each path during the expansion process. Hence, a solution algorithm must expand paths rather than vertices. In contrast to the static algorithm of Dijkstra, which only needs to remember the direct predecessor of each vertex, this must be considered as a severe drawback. The following result shows, that the number of predecessors which are relevant for a further expansion of a path is bounded.

Lemma 4 *Consider a dynamical network in which Assumption 2 holds, and let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given. Let d denote the minimum-hop distance from v_0 to v', and suppose that γ is either linear or logarithmic. Then the number N of predecessors, relevant for the expansion of any path, is bounded by $N \leq \gamma(d\overline{\tau})/\underline{\tau} - 1$.*

Proof: Without loss of generality, we assume that $s_0 = 0$. Let $p_K \in \hat{\mathcal{P}}$ be any feasible path of maximum length $K \in \mathbb{N}$. (Note, that as a consequence of Theorem 2, the length of any feasible path is bounded.) Let $\epsilon_K = \Pi_2(p_K) = (e_1, ..., e_K)$ denote the corresponding topological path, and let further $v_k = \alpha(e_k)$, $\epsilon_k = (e_1, ..., e_k)$ and $p_k = \Pi^{-1}(0, \epsilon_k)$ for $k = 1, ..., K$. For each v_i, $i = 1, ..., k$, the set of feasible visiting times obviously satisfies $S(v'; v_i) \subset [0, \Gamma(t(p_i))]$, as Γ is monotone increasing and $t(p_i) \geq t^*(v_i; v_0, 0) \geq 0$. A necessary condition for the relevance of v_i for the further extension of p_k is therefore

$$t(p_k) + \underline{\tau} \leq \Gamma(t(p_i)), \tag{29}$$

because v_i must still be reachable, and $t(p_{k+1}) \geq t(p_k) + \underline{\tau}$.
Let $p_{i,k} = \Pi^{-1}(t(p_i), (e_{i+1}, ..., e_k))$, $1 \leq i < k \leq K$, denote the tail path of p_k emanating from v_i. Since $t(p_k) = t(p_i) + t(p_{i,k})$ and $\Gamma(t(p_i)) = t(p_i) + \gamma(t(p_i))$, (29) implies, that

$$t(p_{i,k}) + \underline{\tau} \leq \gamma(t(p_i)) \tag{30}$$

is necessary for the relevance of v_i. As $t^*(v';v_0,0) \le d\overline{\tau}$, and v' must always be reachable, another necessary condition for the further extension of p_k is given by

$$t(p_k) + \underline{\tau} \le \Gamma(d\overline{\tau}). \tag{31}$$

Let $j = k - i$ denote the number of relevant predecessors of a path of length k, $\tau_i = t(p_i)/i$ the average edge travel time on p_i and $\tau_j = t(p_{i,k})/j$ the average edge travel time on $p_{i,k}$. (26) implies that $\underline{\tau} \le \tau_i \le \overline{\tau}$ and $\underline{\tau} \le \tau_j \le \overline{\tau}$. We now consider the following nonlinear optimization problem:

$$\min_{(i,j,\tau_i,\tau_j)} -j, \tag{32}$$
$$-i \le 0, \tag{33}$$
$$-j \le 0, \tag{34}$$
$$\underline{\tau} - \tau_i \le 0, \tag{35}$$
$$\tau_i - \overline{\tau} \le 0, \tag{36}$$
$$\underline{\tau} - \tau_j \le 0, \tag{37}$$
$$\tau_j - \overline{\tau} \le 0, \tag{38}$$
$$-\gamma(\tau_i i) + \tau_j j + \underline{\tau} \le 0, \tag{39}$$
$$-\Gamma(\overline{\tau}d) + \tau_j j + \tau_i i + \underline{\tau} \le 0. \tag{40}$$

The constraints (33), (34) ensure, that only paths of nonnegative length are considered. (35)-(38) denote the edge travel time constraints, and (39), (40) coincide with (30), (31). If $x^* = (i^*, j^*, \tau_i^*, \tau_j^*)$ is an optimal solution of (32)-(40), then the number of relevant predecessors is bounded from above by j^*. Let $f : \mathbb{R}^4 \to \mathbb{R}$ denote the objective function of (32), and let $q : \mathbb{R}^4 \to \mathbb{R}$, with the components q_l, $l = 1, ..., 8$, be defined by (33)-(40). According to [5, Theorem 3.3.5], a necessary condition for the optimality of x^* is the existence of $\mu_l \in \mathbb{R}$, $\mu_l \le 0$, $l = 1, ..., 8$, such that

$$-\nabla f(x^*) + \sum_{l=1}^{8} \mu_l \nabla q_l(x^*) = 0, \tag{41}$$
$$\mu_l q_l(x^*) = 0, \quad l = 1, ..., 8, \tag{42}$$

if the set $\Omega = \{x \in \mathbb{R}^4 : q(x) \le 0\}$ satisfies the constaint qualification [5, Definition 3.3.1] in x^*. This is guaranteed by the existence of $\delta x \in \mathbb{R}^4$ with

$$\langle \nabla q_l(x^*), \delta x\rangle < 0 \quad \forall l \in \{1, ..., 8\} \text{ with } q_l(x^*) = 0. \tag{43}$$

according to [5, Theorem 3.3.21].

If $\gamma \equiv \gamma_{\text{lin}}$, an analysis of (41) and (42) yields the admissible solutions $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_6 = 0$, $\mu_5 = -(d\overline{\tau} - \underline{\tau})/\underline{\tau}^2$, $\mu_7 = \mu_8 = -1/2\underline{\tau}$, $i^* = d\overline{\tau}/\tau_i^*$, $j^* = (d\overline{\tau} - \underline{\tau})/\underline{\tau}$, $\tau_j^* = \underline{\tau}$ and $\tau_i^* \in [\underline{\tau}, \overline{\tau}]$ arbitrary. Obviously, the choice of τ_i^* does not affect the value of the objective function. We therefore choose

$\tau_i^* = \overline{\tau}$ and $i^* = d$ as candidates for an optimal solution. The constraint qualification is satisfied in the thereby defined point $x^* = (i^*, j^*, \tau_i^*, \tau_j^*)$, as $\delta x = (0, -3d/\underline{\tau}, -1, d\underline{\tau}/(d\overline{\tau} - \underline{\tau}))$ satisfies (43). Hence the number of relevant predecessors is bounded from above by $j^* = \gamma_{\text{lin}}(d\overline{\tau})/\underline{\tau} - 1$, if $\gamma \equiv \gamma_{\text{lin}}$.
If $\gamma \equiv \gamma_{\log}$, an analysis of (41) and (42) yields the (unique) admissible solution $\mu_1 = \mu_2 = \mu_3 = \mu_6 = 0$, $\mu_4 = -(d - \underline{\tau})/[(1 + d\overline{\tau})\underline{\tau}]$, $\mu_5 = (-\log(\underline{\tau}) + \underline{\tau})/\underline{\tau}$, $\mu_7 = -d\overline{\tau}/\underline{\tau}(1 + d\overline{\tau})$, $\mu_8 = -1/\underline{\tau}(1 + d\overline{\tau})$, $i^* = d$,$j^* = (\log(d\overline{\tau}) - \underline{\tau})/\underline{\tau}$, $\tau_i^* = \overline{\tau}$, $\tau_j^* = \underline{\tau}$. The constraint qualification is satisfied in the thereby defined point $x^* = (i^*, j^*, \tau_i^*, \tau_j^*)$, as $\delta x = (0, -2d(\overline{\tau} + 1)/(\underline{\tau}\overline{\tau}), -1, \underline{\tau}/(\log(d\overline{\tau}) - \underline{\tau}))$ satisfies (43). Hence the number of relevant predecessors is bounded from above by $j^* = \gamma_{\log}(d\overline{\tau})/\underline{\tau} - 1$, if $\gamma \equiv \gamma_{\log}$. □

Remark 13 *Note that the upper bound on the number of predecessors given by Lemma 4 is valid for any feasible path in the dynamical network, given a source vertex and a goal vertex of minimum-hop distance d. In the same manner, in which this bound was derived in the proof of Lemma 4, replacing d by k, a bound for any feasible path of length $k \in \mathbb{N}$ can be derived. This bound will be much smaller for a (topologically) short path, but it will be valid only for any path of length k.*

5 Complexity Results

We have derived two pruning techniques in the last Section, which allow a significant reduction of the cost of computing dynamical optimal paths. Nevertheless, the computation of such paths is still in general NP-hard. In this section, we will prove new complexity results for the computation of time-constrained dynamical optimal paths. As those results will be based on the knowledge of the feasible time intervals, the first result concerns the computation of the feasible visiting times.

Corollary 2 *Consider a dynamical network in which Assumption 2 holds, and let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given. Then the computation of the feasible visiting time intervals $S(v'; v)$ can be carried out in $\mathcal{O}(n \log(n) + m)$ complexity.*

Proof: The computation of the feasible visiting times consists of the computation of two bounds, i.e., the computation of the lower bound $t^*(v; v_0, s_0)$ for each $v \in V$ and the computation of the last visiting time, which allows an arrival at v' at time $s \leq \Gamma(t^*(v'; v_0, s_0))$. The computation of the earliest arrival times can be carried out in $\mathcal{O}(n \log(n) + m)$ complexity, due to Lemma 1. The same holds for the computation of the last departure times, which has been shown in [8]. □

There has been considerable effort in bounding the number of vertices expanded by heuristic search algorithms, such as the A*-algorithm ([18]), in terms of the accuracy of the heuristic. Assuming that the graph is a tree, it has been shown that the number of vertices expanded by the A*-algorithm is polynomial in the

length of the optimal solution (in the worst case), if the accuracy of the heuristic is constant ([26]) or logarithmic ([25]). By contrast, the number of vertices expanded by the A*-algorithm is exponential (in the worst case), if the accuracy of the heuristic is linear ([27]). Although the setting considered in these works does not carry over to the time-dependent case, a similar result holds, if the time variable is discrete and time constraints of varying order are considered.
As we have argued in Section 4, the constraint of allowing only simple paths for expansion leads to a different notion of expansion. In constrast to the usual optimal path algorithms (such as Dijkstra or Bellman-Ford), it is necessary to expand paths rather than nodes. As the number of simple paths grows exponentially with the number of feasible vertices, we cannot expect a polynomial bound on the number of paths. Hence, as long as we consider a discrete time variable, we will only impose a constraint on the visiting times of a vertex, but we will not require paths to be simple.

Theorem 3 *Let $(V, E, \tau; \beta)$ be a dynamical network satisfying Assumption 2, and let $\tau(E \times \mathbb{R}) \subseteq \{\underline{\tau}, ..., \overline{\tau}\}$ with $\underline{\tau}, \overline{\tau} \in \mathbb{N}$. Let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given and let d denote the minimum-hop distance from v_0 to v'. If (V, E) is a symmetric directed r-ary tree, then the number N of feasible vertices in the time-expanded network is*

$$N = \mathcal{O}\left(d^3 r^{d\overline{\tau}/(2\underline{\tau})}\right), \text{ if } \gamma \equiv \gamma_{lin}, \tag{44}$$

$$N = \mathcal{O}\left(d^{1+1/(2\underline{\tau})} \log(d) r^{1/(2\underline{\tau})}\right), \text{ if } \gamma \equiv \gamma_{\log}. \tag{45}$$

Proof: The fastest path subtree T of (V, E) is a directed tree rooted in v_0. As any admissible path must visit v_0 at a feasible time $s \in S(v'; v_0) = \{s_0\}$, the only edge emanating from v_0 must be an edge on a fastest path from v_0 to v'. Due to the FIFO-condition, the fastest path from v_0 to v' is simple and therefore uniquely determined. We denote the vertices, which are passed by this path by $v_0, v_1, ..., v_{d-1}, v_d$, with $v_d = v'$. Let T_k, $k = 1, ..., d$, denote the subtree of T rooted in v_k and containing (except for v_k) only vertices not passed by the fastest path from v_0 to v' (see Figure 3). The number of feasible vertices in the time-expanded network is given by the set of all vertex-time pairs in the time-expansions of the subtrees T_k, $k = 1, ..., d$. As γ is monotonically increasing, the maximum number of feasible copies of v_k is given by $\lfloor \gamma(k\overline{\tau}) \rfloor$. The maximum depth of T_k is therefore bounded from above by $\lfloor \gamma(k\overline{\tau})/(2\underline{\tau}) \rfloor$, because v_k must be reachable at a feasible visiting time from any vertex $v \in T_k$. Moreover, if we consider a vertex v_{kj} at depth $j \in \mathbb{N}$ in T_k (see Figure 3), then $S(v'; v_{kj})$ contains no more than $\gamma(k\overline{\tau}) - 2j\underline{\tau}$ feasible visiting times. The number N_k of feasible vertex-time pairs in the time-expansion of T_k is therefore bounded by

$$N_k \leq \gamma(k\overline{\tau}) + (r - 1) \sum_{j=1}^{\lfloor \gamma(k\overline{\tau})/(2\underline{\tau}) \rfloor} r^{j-1} (\gamma(k\overline{\tau}) - 2j\underline{\tau}). \tag{46}$$

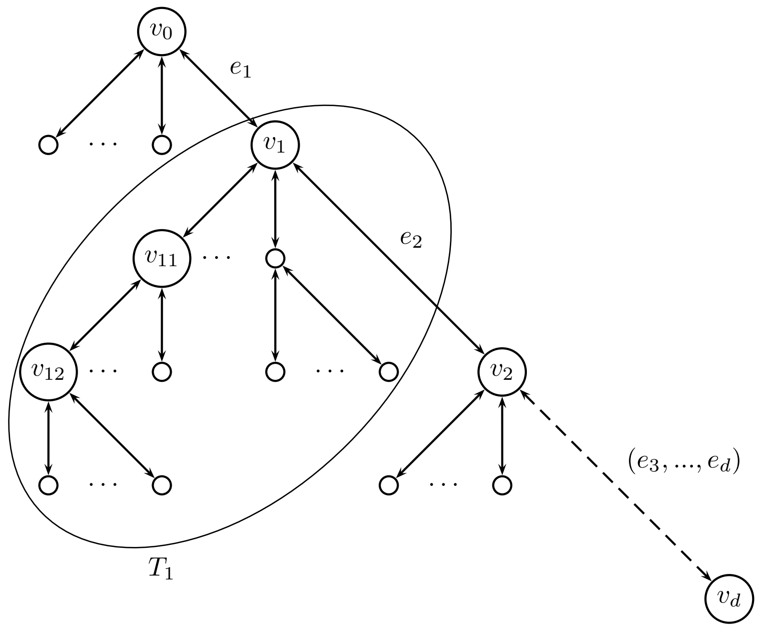

Figure 3: Labelling of the symmetric directed r-ary tree used in the proof of Theorem 3. The edge sequence $(e_1, ..., e_d)$ constitutes the topological structure of the optimal path.

From our reasoning above we have $N \leq \sum_{k=1}^{d} N_k$.
If $\gamma \equiv \gamma_{\text{lin}}$, then (46) becomes

$$N_k \leq k\overline{\tau} + (r-1) \sum_{j=1}^{\lfloor (k\overline{\tau})/(2\underline{\tau}) \rfloor} r^{j-1}(k\overline{\tau} - 2j\underline{\tau}) = \mathcal{O}\left(k^2 r^{k\overline{\tau}/(2\underline{\tau})}\right),$$

which results in (44).
If $\gamma \equiv \gamma_{\log}$, using the formula for the geometric series, (46) becomes

$$\begin{aligned} N_k &\leq \log(k\overline{\tau}) + (r-1) \sum_{j=1}^{\lfloor \log(k\overline{\tau})/(2\underline{\tau}) \rfloor} r^{j-1}(\log(k\overline{\tau}) - 2j\underline{\tau}) \\ &\leq \log(k\overline{\tau}) \left(1 + (r-1) \sum_{j=0}^{\lfloor \log(k\overline{\tau})/(2\underline{\tau}) \rfloor - 1} r^j \right) \\ &= \log(k\overline{\tau}) \left(1 + (r-1) \frac{r^{\lfloor \log(k\overline{\tau})/(2\underline{\tau}) \rfloor} - 1}{r-1}\right) \\ &= \mathcal{O}\left(\log(k)(kr)^{1/(2\underline{\tau})}\right), \end{aligned}$$

which results in (45). □

A major difficulty when adapting this methodology to general graphs is the fact, that there exists more than one simple solution path. In a grid graph, which may be considered as an appropriate model for the road network of an urban area, neither the complexity results concerning the accuracy of a heuristic, nor the results derived in Theorem 3 apply. Considering a continuous variable, independent of the simple path constraint, even the following negative result holds.

Theorem 4 *Let $(V, E, \tau; \beta)$ be a dynamical network satisfying Assumption 2 and suppose that (V, E) is a grid graph. Let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given and let d denote the minimum-hop distance from v_0 to v'. If $\gamma \not\equiv 0$, then in the worst case there exist $\Omega(2^{d/2})$ optimal paths from v to v' and $\Omega(2^{d/2})$ feasible vertex-time pairs.*

Proof: In order to localize a vertex in the grid graph, we use a coordinate system and choose v_0 as the origin. The coordinates $(x, y) \in \mathbb{Z}^2$ of any vertex $v \in V$ in the grid graph are then given by the (directed) number of hops x in the horizontal direction and the (directed) number of hops y in the vertical direction, which are required to reach v from v_0. Without loss of generality, we assume that v' is located at $(x', y') \in \mathbb{Z}^2$, with $0 \le x' \le y'$, $d = x' + y'$. We will now consider the set $V_\Box$ of vertices v with coordinates $(x, y) \in \mathbb{Z}^2$, $0 \le x \le x'$, $0 \le y \le y'$, i.e., those vertices which are contained in minimum-hop paths from v_0 to v'. As $t^*(v; v_0, s_0) \ge \underline{\tau} > 0$ and $\gamma \not\equiv 0$, $S(v'; v)$ contains an infinite number of time instances for all $v \in V_\Box$, $v \ne v_0$. We may therefore choose the edge travel times τ such that each minimum-hop path from v_0 to $v \in V_\Box$ is feasible, and such that each minimum-hop path defines a different arrival time. Furthermore, we may choose the edge cost β, such that $\beta(e, s) = \underline{\beta} > 0$ for all $s \in S(v'; \alpha(e))$ and all $e \in E$ with $e = (u, v)$ for some $u, v \in V_\Box$, and $\beta(e, s) > \underline{\beta}$ otherwise. With this choice, each minimum-hop path from v_0 to v' is feasible and optimal. Each of these paths can be represented by a sequence of x' horizontal and y' vertical hops, hence the number of all minimum-hop paths from v_0 to v' is given by the number of permutations of a set containing x' indistintinguishable elements of one type (horizontal hops) and y' indistintinguishable elements of another type (vertical hops). Therefore, there are

$$\frac{(x' + y')!}{x'!y'!} \tag{47}$$

minimum-hop paths. Choosing, without loss of generality, $x' = y' = d/2$, we obtain $\Omega(2^{d/2})$ optimal paths from v to v'. □

Remark 14 *Note, that the exponential number of feasible vertex-time pairs results from the fact, that each dynamical path eventually defines a new vertex-time pair. In this case, it might be beneficial to introduce the simple path constraint, as the number of simple paths of length l in a grid graph is μ^l, $2.62002 \le \mu \le 2.67919$ ([21]), whereas the number of paths of length l is of the order 4^l. Although this may lead to a considerable decrease in the number of vertex-time pairs, exponential worst-case complexity can only be avoided by choosing $\gamma \equiv 0$.*

Despite the negative result given by Theorem 4, the number of feasible vertex-time pairs in a time-dependent grid graph remains polynomial in the minimum-hop distance of the source and goal vertex, if the time variable is discrete. In order to establish this result, we need the following Lemma:

Lemma 5 *Let (V, E) be a grid graph. The number of vertices $v \in V$ of minimum-hop distance k from a given vertex v_0 is bounded from above by $4k$.*

Proof: Associating the same coordinate system with the grid graph as in the proof of Theorem 4, the number of vertices of distance k is given by the number of solutions $(i, j) \in \mathbb{Z}^2$ of $|i|+|j| = k$. These solutions form a $\pi/4$-rotated square in $\mathbb{Z}^2$, with each edge of the square containing $k+1$ grid points. As each corner of the square is contained in two edges, there are $4(k+1) - 4 = 4k$ vertices of minimum-hop distance k from v_0. □

We now derive an upper bound for the number of vertex-time pairs, which implies the desired complexity result for discrete-time time-expanded grid graphs.

Theorem 5 *Let $(V, E, \tau; \beta)$ be a dynamical network satisfying Assumption 2, and let $\tau(E \times \mathbb{R}) = \{\underline{\tau}, ..., \overline{\tau}\}$ with $\underline{\tau}, \overline{\tau} \in \mathbb{N}$. Let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given and let d denote the minimum-hop distance from v_0 to v'. Suppose that the number of neighbours of minimum-hop distance k from v_0 is bounded by $\nu(k)$. Then the number N of feasible vertices in the time-expanded network is bounded by*

$$N \leq \sum_{k=1}^{\lfloor \Gamma(d\overline{\tau})/\underline{\tau} \rfloor} \nu(k)\gamma(k\overline{\tau}). \tag{48}$$

Proof: Let v_k denote a vertex of minimum-hop distance k from the source vertex v_0, and let $t_k = t^*(v_k; v_0, s_0)$. From (26), we deduce that $k\underline{\tau} \leq t_k \leq k\overline{\tau}$, and $t^*(v'; v_0, s_0) \leq d\overline{\tau}$. Relaxing the constraint (10), which ensures that v' can be reached at a feasible time from each $s \in S(v'; v_k)$, $S(v'; v_k)$ contains at most $\lfloor \gamma(t_k) \rfloor$ feasible passing times, and t_k is bounded from above by $\overline{t} = \Gamma(d\overline{\tau})$. An upper bound for the number of feasible vertex-time pairs of minimum-hop distance at most L from v_0 is therefore given by the following optimization problem:

$$\max_{(t_1,...,t_L)} \sum_{k=1}^{L} \nu(k)\gamma(t_k), \tag{49}$$

$$\underline{\tau} \leq t_1 \leq \overline{\tau}, \tag{50}$$

$$\underline{\tau} \leq t_{k+1} - t_k \leq \overline{\tau}, \qquad k = 1, ..., L-1, \tag{51}$$

$$t_k \leq \overline{t}, \qquad k = 1, ..., L. \tag{52}$$

The constraints (50) and (51) ensure, that the bound on the edge travel times (26) is satisfied. Obviously, all t_k, $k = 1, ..., L$, are bounded from above by $L\overline{\tau}$, hence $\sum_{k=1}^{L} \nu(k)\gamma(t_k)$ is bounded from above, and if there exists a solution, there also exists an optimal solution with a finite value N_L of the objective function (49). As we have required $t_k \leq \overline{t}$ for all $k = 1, ..., L$, a solution can only exist if $L \leq \overline{t}/\underline{\tau}$. Hence, the number of feasible vertex-time pairs is bounded by

$$N \leq \max_{L \in \{1,...,\lfloor \overline{t}/\underline{\tau} \rfloor\}} N_L. \tag{53}$$

Since γ is monotone increasing, for any $L \in \{1, ..., \lfloor \overline{t}/\underline{\tau} \rfloor\}$, $\sum_{k=1}^{L} \nu(k)\gamma(t_k)$ is maximized if the variables t_k are maximized simultaneously, i.e., if for some $k^* \in \{1, ..., L\}$

$$t_k = \overline{t} - (L-k)\underline{\tau}, \qquad k^* + 1 \leq k \leq L, \tag{54}$$

$$t_{k^*} = \overline{t} - (L - k^* + 1)\underline{\tau} - (k^* - 1)\overline{\tau}, \tag{55}$$

$$t_k = k\overline{\tau}, \qquad 1 \leq k \leq k^* - 1. \tag{56}$$

From (54)-(56) we see that $t_k \leq k\overline{\tau}$ for all $k = 1, ..., L$. Consequently, because γ is monotone increasing, we obtain $\gamma(t_k) \leq \gamma(k\overline{\tau})$ and

$$\sum_{k=1}^{L} \nu(k)\gamma(t_k) \leq \sum_{k=1}^{L} \nu(k)\gamma(k\overline{\tau}).$$

Finally, as $L \leq \overline{t}/\underline{\tau} = \Gamma(d\overline{\tau})/\underline{\tau}$, we obtain (48). □

Remark 15 *In the proof of Theorem 5, the optimization problem (49)-(51) defines a bound for the number of feasible vertex-time pairs, which only accounts for the distance to the source vertex v_0. Considering, in addition to (50)-(52), the constraint that v' must be reachable at a feasible passing time from any feasible vertex-time pair, a more sophisticated and more accurate upper bound for the number of feasible vertex-time pairs can be defined as follows: Associate with any $v \in V$ the minimum-hop distance i from v_0 and the minimum-hop distance j from v'. (Note, that we must assume that the number of neighbours of minimum-hop distance j from v' is bounded by $\nu(j)$.) Then, for any $L \in \{1, ..., \lfloor \Gamma(d\overline{\tau})/\underline{\tau} \rfloor\}$, solve the following maximization problem:*

$$\max_{\nu_{ij}, t_{ij}} \sum_{i+j \leq L,\, i,j \geq 0} \nu_{ij}\gamma(t_{ij}), \tag{57}$$

$$i\underline{\tau} \leq t_{ij} \leq i\overline{\tau}, \qquad i+j \leq L,\ i,j \geq 0, \tag{58}$$

$$\Gamma(d\overline{\tau}) - j\overline{\tau} \leq t_{ij} \leq \Gamma(d\overline{\tau}) - j\underline{\tau}, \qquad i+j \leq L,\ i,j \geq 0, \tag{59}$$

$$\nu_{ij} \leq \nu(i), \qquad i+j \leq L,\ i,j \geq 0, \tag{60}$$

$$\nu_{ij} \leq \nu(j), \qquad i+j \leq L,\ i,j \geq 0. \tag{61}$$

In this formulation, (58) and (59) take into account the time constraints at $v \sim (i,j)$, whereas (60) and (61) take into account the topological structure of the dynamical network. The maximum value of the objective function in (57) defines an upper bound for the maximum number of feasible vertex-time pairs.

As long as neither γ nor ν are exponential functions, this procedure only yields a more accurate upper bound, but does not improve the result of Theorem 5 in the order of complexity. For this reason, we have not further followed this approach.

Remark 16 *Note, that the application of Theorem 5 to a symmetrical r-ary tree results in different orders of complexity than Theorem 3, i.e., $N = \mathcal{O}\left(d^2 r^{2d\overline{\tau}/\underline{\tau}}\right)$ if $\gamma \equiv \gamma_{lin}$ and $N = \mathcal{O}\left(d^{1+1/\underline{\tau}} \log(d) r^{d\overline{\tau}/\underline{\tau}+1/\underline{\tau}}\right)$ if $\gamma \equiv \gamma_{\log}$. The fact, that N grows exponentially with d even if $\gamma \equiv \gamma_{\log}$ is due to the weaker structural assumptions in Theorem 5.*

Corollary 3 *Let $(V, E, \tau; \beta)$ be a dynamical network satisfying Assumption 2, and let $\tau(E \times \mathbb{R}) = \{\underline{\tau}, ..., \overline{\tau}\}$ with $\underline{\tau}, \overline{\tau} \in \mathbb{N}$. Let a source vertex $v_0 \in V$, a departure time $s_0 \in \mathbb{R}$ and a goal vertex $v' \in V$ be given and let d denote the minimum-hop distance from v_0 to v'. If (V, E) is a grid graph, then the number N of feasible vertices in the time-expanded network is*

$$N = \mathcal{O}(d^3), \ \ \text{if } \gamma \equiv \gamma_{lin}, \tag{62}$$

$$N = \mathcal{O}(d^2 \log(d)), \ \ \text{if } \gamma \equiv \gamma_{\log}. \tag{63}$$

Proof: The assertion follows directly from Lemma 5 and Theorem 5, since for $\gamma \equiv \gamma_{\text{lin}}$ and $\gamma \equiv \gamma_{\log}$ we have $\gamma(k\overline{\tau}) = \mathcal{O}(\gamma(k))$ and $\Gamma(d\overline{\tau}/\underline{\tau}) = \mathcal{O}(d)$. $\square$

6 Conclusion

In this paper, we have considered the problem of computing cost-optimal paths in time-dependent networks. We have considered a topological and a time constraint, which induce, that cost-optimal paths stay close to fastest paths. Assuming that the dynamical network satisfies the FIFO-condition, we have shown that the time constraint can be guaranteed in polynomial time. We have derived new pruning criteria, one of which is applicable in both the constrained and the unconstrained setting of the cost-optimal dynamical path problem. We have proved, that there is no time constraint, except the constraint of allowing only fastest paths, which results in a polynomial complexity bound in continuous-time grid graphs. Assuming a discrete time variable, we have shown, that the number of feasible vertex-time pairs in a time-expanded r-ary tree is exponential in the length of the solution path, if a linear time constraint is applied, whereas this number is polynomial in the length of the solution path, if a logarithmic time constraint is applied. Moreover, we have proved, that the number of feasible vertex-time pairs is polynomial in a discrete-time time-expanded grid graph for both a linear and a logarithmic time constraint.

A direction of further research could be the investigation of the effect of time constraints in dynamical networks with other topological structures than the ones considered in this paper. Another possibility would be to study the effect of the accuracy of the heuristic used by heuristic search algorithms in dynamical networks. In view of possible applications, it would also be of interest to carry out a detailed empirical study of the effects of the proposed constraints and pruning criteria, e.g., in a large dynamical road network.

References

[1] A. Ahuja, J. Orlin, S. Pallotino, and M. Scutella. Dynamic shortest paths minimizing travel times and cost. *MIT Sloan Working Paper*, (4390-02), August 2002.

[2] R. Ahuja, T. Magnati, and J. Orlin. *Network flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[3] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[4] X. Cai, T. Kloks, and C. Wong. Time-varying shortest path problems with constraints. *Networks*, 29:141–149, 1997.

[5] M. Canon, C. Cullum, and E. Polak. *Theory of Optimal Control and Mathematical Programming.* McGraw-Hill, 1970.

[6] I. Chabini. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Records*, 1645:170–175, 1998.

[7] K. Cooke and E. Halsey. The shortest route through a network with time dependent internodal transit times. *J. Math. Anal. Appl.*, 14:493–498, 1966.

[8] C. Daganzo. Reversibility of the time-dependent shortest path problem. Technical report, Institute of Transportation Studies, 1998.

[9] B. Dean. Continuous-time dynamic shortest path algorithms. Master's thesis, MIT, 1999.

[10] B. Dean. Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks*, 33(1):41–46, 2004.

[11] D. Delling and D. Wagner. Time-Dependent Route Planning. In R. K. Ahuja, R. H. Möhring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, Lecture Notes in Computer Science. Springer, 2009. accepted for publication, to appear.

[12] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[13] L. Ford Jr. and D. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433, 1958.

[14] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of ACM*, 34:596–615, 1987.

[15] L. Fu and L. Rilett. Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research, Part B*, 1998.

[16] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks part i: Framework and taxonomy. pages 549–554, Singapore, September 2002. The IEEE 5th International Conference on Intelligent Transportation Systems.

[17] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks part ii: Exact and approximation algorithms. pages 555–559, Singapore, September 2002. The IEEE 5th International Conference on Intelligent Transportation Systems.

[18] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of systems science and cybernetics*, SSC-4(2):100–107, July 1968.

[19] O. Junge and H. Osinga. A set oriented approach to global optimal control. *ESAIM: Control, Optimisation and Calculus of Variations*, 10:259–270, 2004.

[20] S. Kim, M. Lewis, and C. White. Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):178–188, June 2005.

[21] M. Liskiewicz, O. Mitsunori, and S. Toda. The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes. *Theoretical Computer Science*, 304:129–156, 2003.

[22] D. Medhi and K. Ramasamy. *Network Routing. Algorithms, Protocols, and Architectures*. Morgan Kaufmann, 2007.

[23] A. Orda and R. Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the Association for Computing Machinery*, 37:607–625, July 1990.

[24] A. Orda and R. Rom. Minimum weight paths in time-dependent networks. *Networks*, 21:295–319, 1991.

[25] J. Pearl. *Heuristics - Intelligent Search Stategies for Computer Problem Solving*. Addison-Wesley, 1984.

[26] I. Pohl. First results on the effect of error in heuristic search. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 5, pages 219–236. University Press, 1969.

[27] I. Pohl. Practical and theoretical considerations in heuristic search algorithms. In E. W. Elcock and D. Michie, editors, *Machine Intelligence*, volume 9, pages 55–72. 1977.

[28] K. Sung, M. Bell, S. Myeongki, and S. Park. Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research*, 121:32–39, 2000.