



UNIVERSITÀ DI PARMA

ARCHIVIO DELLA RICERCA

University of Parma Research Repository

Solution algorithms for the Bounded Acceleration Shortest Path Problem

This is the peer reviewed version of the following article:

Original

Solution algorithms for the Bounded Acceleration Shortest Path Problem / Ardizzoni, S., Consolini, L., Laurini, M., Locatelli, M.. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - 68:3(2023), pp. 1910-1917. [10.1109/TAC.2022.3172169]

Availability:

This version is available at: 11381/2932182 since: 2023-12-07T15:57:44Z

Publisher:

Published

DOI:10.1109/TAC.2022.3172169

Terms of use:

Anyone can freely access the full text of works made available as "Open Access". Works made available

Publisher copyright

note finali coverpage

(Article begins on next page)

12 June 2026

GENERAL INSTRUCTION

- **Authors:** Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections.
- **Authors:** We cannot accept new source files as corrections for your article. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

QUERIES

- Q1. Author: Please supply index terms/keywords for this article. To download the IEEE Taxonomy go to https://www.ieee.org/documents/taxonomy_v101.pdf.
- Q2. Author: Please confirm or add details for any funding or financial support for the research of this article.
- Q3. Author: Please provide the report number in Ref. [4].
- Q4. Author: Please update Ref. [17]
- Q5. Author: Please provide the location of the publisher in Ref. [19].

Technical Notes and Correspondence

Solution Algorithms for the Bounded Acceleration Shortest Path Problem

Stefano Ardizzoni , Luca Consolini , Mattia Laurini , and Marco Locatelli 

Abstract—The purpose of this article is to introduce and characterize the bounded acceleration shortest path problem (BASP), a generalization of the shortest path problem (SP). This problem is associated to a graph: nodes represent positions of a mobile vehicle and arcs are associated to preassigned geometric paths that connect these positions. The BASP consists in finding the minimum-time path between two nodes. Differently from the SP, the vehicle has to satisfy bounds on maximum and minimum acceleration and speed, which depend on the vehicle's position along the currently traveled arc. Even if the BASP is NP-hard in the general case, we present a solution algorithm that achieves polynomial time-complexity under some additional hypotheses on problem data.

Index Terms—

I. INTRODUCTION

The combinatorial problem of detecting the best path from a source to a destination node over an oriented graph with constant costs associated to its arcs, also known as shortest path problem (SP in what follows), is well known and can be efficiently solved, e.g., by the Dijkstra algorithm (in case of nonnegative costs). The continuous problem of minimum-time speed planning over a fixed path under given speed and acceleration constraints, also depending on the position along the path, is also widely studied and very efficient algorithms for its solution exist. But the combination of these two problems, called in what follows bounded acceleration shortest path problem (BASP), turns out to be more challenging than the two problems considered separately. More precisely, in terms of the complexity theory, it is possible to prove that the BASP is NP-hard, while the two problems considered separately are both polynomially solvable. In the BASP, we still have the combinatorial search for a best path as in SP but, differently from SP, the cost of an arc (more precisely, the time to traverse it) is not a constant value but depends on the speed planning along the arc itself, which, in turn, depends on the speed and acceleration constraints not only over the same arc but also over those preceding and following it in the selected path. Fig. 1(a) presents a simple scenario that allows

Manuscript received October 25, 2021; accepted April 15, 2022. This work was supported by the Programme "FIL-Quota Incentivante" of University of Parma and co-sponsored by Fondazione Cariparma. Recommended by Associate Editor T. Faulwasser. (Corresponding author: Luca Consolini.)

The authors are with the Dipartimento di Ingegneria e Architettura, Università degli Studi di Parma, 43124 Parma, Italy (e-mail: stefano.ardizzoni@unipr.it; luca.consolini@unipr.it; mattia.laurini@unipr.it; marco.locatelli@unipr.it).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3172169>.

Digital Object Identifier 10.1109/TAC.2022.3172169

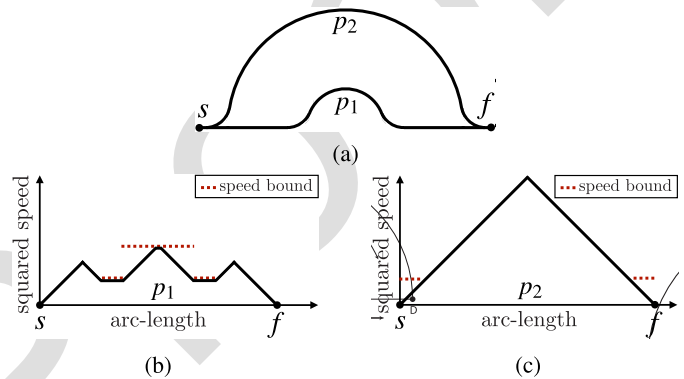


Fig. 1. Comparison of BASP and SP solutions. (a) Paths p_1 and p_2 connecting node s and f . (b) Optimal speed profile on p_1 . (c) Optimal speed profile on p_2 .

to illustrate the BASP and its difference with SP; it shows two fixed paths p_1 and p_2 connecting positions s and f . The vehicle starts from s with null speed and must reach f with null speed. The solution of SP corresponds to the path p_1 , which is the one of the shortest length. The BASP consists in finding the shortest-time path under acceleration and speed constraints. In this case, we assume that the vehicle acceleration and deceleration are bounded by a common constant and that its speed is bounded only on the central, high-curvature section of p_1 , in order to avoid excessive lateral acceleration, which may cause sideslip. If the bound on acceleration and deceleration is sufficiently high, the solution of the BASP corresponds to the path p_2 . Indeed, even if the latter path is longer, it can be traveled with a greater mean speed. Fig. 1(b) represents the fastest speed profile on p_1 . The x -axis corresponds to the arc-length position on the path p_1 and the y -axis represents the squared speed. In this representation, arc-length intervals of constant acceleration or deceleration correspond to straight lines. Fig. 1(c) represents the fastest speed profile on p_2 . Even if path p_2 is longer than p_1 , it can be traveled in less time. In fact, the vehicle is able to accelerate till the midpoint, and then, to decelerate to the end position f .

The interest for the BASP comes from a specific industrial application, namely the optimization of automated guided vehicles (AGVs) motion in automated warehouses. The AGVs may be either free to move within a facility or be only allowed to move along predetermined paths. In the first case, one needs to employ environmental representations such as cell decomposition methods [1] or trajectory maps [2]. In particular, the authors in [3] present an algorithm based on a modification of Dijkstra's algorithm in which edge weights are history dependent. Our work is related to the second approach. Namely, we assume that AGVs

cannot move freely within their environment and are instead required to move along predetermined paths that connect fixed operating points. These may be associated to shelves locations, where packages are stored or retrieved, to the end of production lines, where AGVs pick up final products, and to additional intermediate locations, used for routing. All these points are formally represented as nodes of a graph, whose arcs represent connecting paths. If AGVs are not subject to acceleration and speed constraints, the minimum-time planning problem is equivalent to SP and can be solved by the Dijkstra algorithm or its variants: see, for instance, [4]–[6], or other algorithms such as A* [7], Lifelong planning A* [8], D* [9], and D* Lite [10]. However, since the motion of AGVs must satisfy constraints on maximum speed and tangential and transversal accelerations that depend on the vehicle position on the path, these approaches cannot be applied to solve the BASP.

Instead, various works consider the minimum-time speed planning problem with acceleration and speed constraint on an *assigned* path. For instance, one can use the methods presented in [11] and [12], or path-following techniques such as [13] and [14].

As said, despite the fact that a large literature exists on SP and on the minimum-time speed planning on an assigned path, to the authors' knowledge, the BASP has never been specifically addressed in the literature. Formally, the BASP can be framed as an optimal control problem for a switching system, in which switchings are associated to passages from arc to arc and each discrete state is associated to a specific set of constraints. The results presented in this article exploit the very specific structure of the BASP and cannot be applied to generic switching systems. Anyway, the Algorithm V.5 could still apply to other switching systems satisfying an analogous of Proposition IV.3 and identifying a class of such systems could be the topic of future research.

This article is structured as follows. After introducing the notation employed throughout this article in Section II, in Section III, we first briefly discuss the solution of the speed planning problem along a *fixed* path, and then, we provide a formal statement of the BASP, also mentioning an NP-hardness result. In Section IV, we consider a subclass of the BASP, called k -BASP, which can be solved with polynomial time complexity for fixed values of k . Since constant k is problem dependent and is not known in advance, in Section V, we present an adaptive A* algorithm to find k . Finally, Section VI presents different computational experiments.

II. NOTATION

A directed graph is a pair $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where \mathbb{V} is a set of nodes and $\mathbb{E} \subset \{(x, y) \in \mathbb{V}^2 \mid x \neq y\}$ is a set of directed arcs. A path p on \mathbb{G} is a sequence of adjacent nodes of \mathbb{V} (i.e., $p = \sigma_1 \cdots \sigma_m$, with $(\forall i \in \{1, \dots, m\}) (\sigma_i, \sigma_{i+1}) \in \mathbb{E}$). An alphabet $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ is a set of symbols. A word is any finite sequence of symbols. The set of all words over Σ is Σ^* , which also contains the empty word ε , while Σ_i represents the set of all words of length up to $i \in \mathbb{N}$, (i.e., words composed of up to i symbols, including ε). Given a word $w \in \Sigma^*$, $|w|$ denote its length. Given a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, we can think of \mathbb{V} as an alphabet so that any path p of \mathbb{G} is a word in \mathbb{V}^* . Given $s, t \in \Sigma^*$, the word obtained by writing t after s is the concatenation of s and t , denoted by $st \in \Sigma^*$; we call t a suffix of st and s a prefix of st . For $r \in \Sigma^*$, \bar{r} is the rightmost symbol of r . In the following, we represent paths of \mathbb{G} as strings of symbols in \mathbb{V} . This allows to use the concatenation operation on paths and to use prefixes and suffixes to represent portions of paths. For $x \in \mathbb{R}$, $\lceil x \rceil = \min\{i \in \mathbb{Z} \mid i \geq x\}$ is the ceiling of x . For $a, b \in \mathbb{R}$, we set $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$, as the minimum and maximum operations, respectively.

Finally, given an interval $I \subseteq \mathbb{R}$, we recall that $W^{1,\infty}(I)$ is the Sobolev space of functions in $L^\infty(I)$ with weak derivative of order 1 with finite L^∞ -norm. For $f, g \in W^{1,\infty}(I)$, we denote with $f \wedge g$ and $f \vee g$ the point-wise minimum and maximum of f and g , respectively.

III. PROBLEM FORMULATION

Before giving the formal description of the BASP, in Section III-A, we briefly discuss the solution of the speed planning problem along a fixed path. Although such problem has been already widely discussed in the literature, here, we briefly describe a way to tackle it in order to better understand the following formulation of the BASP.

A. Speed Planning Along an Assigned Path

Let $\gamma : [0, \lambda_f] \rightarrow \mathbb{R}^2$ be a C^2 function such that $(\forall \lambda \in [0, \lambda_f]) \|\gamma'(\lambda)\| = 1$. The image set $\gamma([0, \lambda_f])$ represents the path followed by a vehicle, $\gamma(0)$ the initial configuration, and $\gamma(\lambda_f)$ the final one. The function γ is an arc-length parameterization of a path. We want to compute the speed law that minimizes the overall travel time while satisfying some kinematic and dynamic constraints. To this end, let $\xi : [0, t_f] \rightarrow [0, \lambda_f]$ be a differentiable monotonically increasing function representing the vehicle arc-length coordinate along the path as a function of time and let $v : [0, \lambda_f] \rightarrow [0, +\infty)$ be such that $(\forall t \in [0, t_f]) \dot{\xi}(t) = v(\xi(t))$. In this way, $v(\lambda)$ is the vehicle speed at position λ . The vehicle position as a function of time is given by $x : [0, t_f] \rightarrow \mathbb{R}^2$, $x(t) = \gamma(\xi(t))$, speed and acceleration are given by $\dot{x}(t) = \gamma'(\xi(t))v(\xi(t))$, and $\ddot{x}(t) = a_L(t)\gamma'(\xi(t)) + a_N(t)\gamma'^{\perp}(\xi(t))$, where $a_L(t) = v'(\xi(t))v(\xi(t))$ and $a_N(t) = \kappa(\xi(t))v(\xi(t))^2$ are the longitudinal and normal components of acceleration, respectively. Here, $\kappa : [0, \lambda_f] \rightarrow \mathbb{R}$ is the scalar curvature, defined as $\kappa(\lambda) = \langle \gamma''(\lambda), \gamma'(\lambda)^\perp \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the scalar product.

We require to travel distance λ_f in a minimum time while satisfying, for every $t \in [0, \xi^{-1}(\lambda_f)]$, $0 \leq v^-(\xi(t)) \leq v(\xi(t)) \leq v^+(\xi(t))$, $|a_N(\xi(t))| \leq \beta(\xi(t))$, $\alpha^-(\xi(t)) \leq a_L(\xi(t)) \leq \alpha^+(\xi(t))$. Here, functions $v^-, v^+, \alpha^-, \alpha^+$, and β are arc-length-dependent bounds on the vehicle speed and on its longitudinal and normal acceleration. It is convenient to make the change of variables $w = v^2$ (see [15]) so that by setting $\Psi(w) = \int_0^{\lambda_f} w(\lambda)^{-\frac{1}{2}} d\lambda$, $\mu^+(\lambda) = v^+(\lambda)^2 \wedge \frac{\beta(\lambda)}{\kappa(\lambda)}$, and $\mu^-(\lambda) = v^-(\lambda)^2$, our problem takes on the following form.

$$\min_{w \in W^{1,\infty}([0, \lambda_f])} \Psi(w) \quad (1a)$$

$$\mu^-(\lambda) \leq w(\lambda) \leq \mu^+(\lambda), \quad \lambda \in [0, \lambda_f] \quad (1b)$$

$$\alpha^-(\lambda) \leq w'(\lambda) \leq \alpha^+(\lambda), \quad \lambda \in [0, \lambda_f] \quad (1c)$$

where $\Psi : W^{1,\infty}([0, \lambda_f]) \rightarrow \mathbb{R}$ is order reversing (i.e., $(\forall x, y \in [0, \lambda_f]) x \geq y \Rightarrow \Psi(x) \leq \Psi(y)$) and $\mu^-, \mu^+, \alpha^-, \alpha^+ \in L^\infty([0, \lambda_f])$ are assigned functions with $\mu^-, \alpha^+ \geq 0$, and $\alpha^- \leq 0$. Initial and final conditions on speed can be included in the definition of functions μ^- and μ^+ . For instance, to set initial condition $w(0) = w_0$, it is sufficient to define $\mu^+(0) = \mu^-(0) = w_0$. In [16], we introduced a subset of $W^{1,\infty}([0, \lambda_f])$, called Q , as a technical requirement and an operator based on the solution of the following differential equations:

$$\begin{cases} F'(\lambda) = \begin{cases} \alpha^+(\lambda) \wedge \mu'(\lambda), & \text{if } F(\lambda) \geq \mu(\lambda) \\ \alpha^+(\lambda), & \text{if } F(\lambda) < \mu(\lambda) \end{cases} \\ F(0) = \mu(0) \end{cases} \quad (2)$$

$$\begin{cases} B'(\lambda) = \begin{cases} \alpha^-(\lambda) \wedge \mu'(\lambda), & \text{if } B(\lambda) \geq \mu(\lambda) \\ \alpha^-(\lambda), & \text{if } B(\lambda) < \mu(\lambda) \end{cases} \\ B(\lambda_f) = \mu(\lambda_f) \end{cases} \quad (3)$$

with $F, B \in Q$, that allows to compute the optimal solution of the Problem (1). In particular, in [16], it is shown that the optimal solution is $F(\mu^+) \wedge B(\mu^+)$. We refer the reader to [16] for a detailed discussion.

B. BASP Problem

In this section, we provide a formal description of the BASP. Let us consider a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, with $\mathbb{V} = \{\sigma_1, \dots, \sigma_N\}$. For each $i \in \{1, \dots, N\}$, the node σ_i represents an operating point $R_i \in \mathbb{R}^2$. In fact, the restriction $R_i \in \mathbb{R}^2$ is not strictly necessary but we imposed it since it holds in the AGV application, which is the main motivation of this work. Each arc $\theta = (\sigma_i, \sigma_j) \in \mathbb{E}$ represents a fixed directed path between two operating points and is associated to an arc-length parameterized path γ_θ of length $\ell(\theta)$, such that $\gamma_\theta(0) = R_i$ and $\gamma_\theta(\ell(\theta)) = R_j$. In the following, we denote the set of all possible paths on \mathbb{G} by P . Similarly, for $s, f \in \mathbb{V}$, we denote by P_s the subset of P consisting in all paths starting from s and by $P_{s,f}$ the subset of P consisting in all paths starting from s and ending in f . We extend this definition to subsets of \mathbb{V} , that is, if $S, F \subset \mathbb{V}$, we denote by $P_{S,F}$ the set of all paths starting from nodes in S and ending in nodes in F . Given a path $p = \sigma_1 \cdots \sigma_m$, its length $\ell(p)$ is defined as the sum of the lengths of its individual arcs, that is, $\ell(p) = \sum_{i=1}^{m-1} \ell(\sigma_i, \sigma_{i+1})$.

To setup our problem, we need to associate four real-valued functions to each edge $\theta \in \mathbb{E}$: the maximum and minimum allowed acceleration and squared speed. The domain of each function is the arc-length coordinate on the path γ_θ . Then, given a specific path p , we are able to define a speed optimization problem of the form (1) by considering the constraints associated to the edges that compose p . We define the set of edge functions as $\mathcal{E} = \{\varphi : \mathbb{E} \times \mathbb{R}^+ \rightarrow \mathbb{R}\}$. If $\varphi \in \mathcal{E}$, $\theta \in \mathbb{E}$, $\lambda \in \mathbb{R}^+$, $\varphi(\theta, \lambda)$ denotes the value of φ on edge θ at position λ . Note that $\varphi(\theta, \lambda)$ will be relevant only for $\lambda \in [0, \ell(\theta)]$. Given a path $p = \sigma_1 \cdots \sigma_m$, we associate to $\varphi \in \mathcal{E}$ a function $\varphi_p : [0, \ell(p)] \rightarrow \mathbb{R}$ in the following way. Define functions $\Theta : [0, \ell(p)] \rightarrow \mathbb{N}$, $\Lambda : [0, \ell(p)] \rightarrow \mathbb{R}$ such that $\Theta(\lambda) = \max\{i \in \mathbb{N} \mid \ell(\sigma_1 \cdots \sigma_i) \leq \lambda\}$ and $\Lambda(\lambda) = \ell(\sigma_1 \cdots \sigma_{\Theta(\lambda)})$. In this way, $\Theta(\lambda)$ is such that $\theta(\lambda) = (\sigma_{\Theta(\lambda)}, \sigma_{\Theta(\lambda)+1})$ is the edge that contains the position at arc length λ along p , and $\Lambda(\lambda)$ is the sum of the lengths of all arcs up to node $\sigma_{\Theta(\lambda)}$ in p . Then, we define $\varphi_p(\lambda) = \varphi(\theta(\lambda), \lambda - \Lambda(\lambda))$.

Given $\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^+, \hat{\alpha}^- \in \mathcal{E}$ and path $p \in P$, let $\mathbb{B} = (\hat{\mu}^-, \hat{\mu}^+, \hat{\alpha}^-, \hat{\alpha}^+)$. Assume $(\forall \theta \in \mathbb{E}) \hat{\mu}^+(\theta, \cdot) \in Q$ and define $T_{\mathbb{B}}(p) = \min_{w \in W^{1,\infty}([0, \ell(p)])} \Psi(w)$, as the solution of the Problem (1) along path p with $\mu^- = \hat{\mu}^-$, $\mu^+ = \hat{\mu}^+$, $\alpha^- = \hat{\alpha}^-$, and $\alpha^+ = \hat{\alpha}^+$. In this way, $T_{\mathbb{B}}(p)$ is the minimum time required to traverse the path p , respecting the speed and acceleration constraints defined in \mathbb{B} . We set $T_{\mathbb{B}}(p) = +\infty$ if the Problem (1) is not feasible.

The following is the main problem discussed in this article.

Problem III.1 (BASP): Given a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, $\mu^+, \mu^-, \alpha^-, \alpha^+ \in \mathcal{E}$, $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+)$, $s \in \mathbb{V}$, and $F \subset \mathbb{V}$, find $\min_{p \in P_{s,F}} T_{\mathbb{B}}(p)$.

In other words, we want to find the path p that minimizes the transfer time between source node s and a destination node in F , taking into account bounds on speed and accelerations on each traversed arc (represented by arc functions $\mu^+, \mu^-, \alpha^-, \alpha^+$). The following properties are a direct consequence of the definition of $T_{\mathbb{B}}(p)$.

Proposition III.2: The following properties hold:

- 1) let $p_1, p_2 \in P$, $p_1 p_2 \in P \Rightarrow T_{\mathbb{B}}(p_1 p_2) \geq T_{\mathbb{B}}(p_1) + T_{\mathbb{B}}(p_2)$;
- 2) if $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+)$, $\hat{\mathbb{B}} = (\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^-, \hat{\alpha}^+)$ are such that $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) [\mu^-(\theta, \lambda), \mu^+(\theta, \lambda)] \subset [\hat{\mu}^-(\theta, \lambda), \hat{\mu}^+(\theta, \lambda)]$

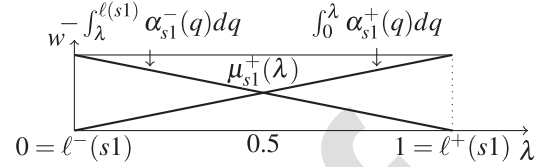


Fig. 2. Computation of $\ell^+(s1) = 1$ and $\ell^-(s1) = 0$.

and $[\alpha^-(\theta, \lambda), \alpha^+(\theta, \lambda)] \subset [\hat{\alpha}^-(\theta, \lambda), \hat{\alpha}^+(\theta, \lambda)]$, then $(\forall p \in P) T_{\mathbb{B}}(p) \geq T_{\hat{\mathbb{B}}}(p)$.

In particular, the first property states that the minimum time for traveling the composite path $p_1 p_2$ is greater or equal to the sum of the times needed for traveling p_1 and p_2 separately. In fact, in the first case, the speed must be continuous when passing from p_1 to p_2 (due to the acceleration bounds), but this constraint does not need to be satisfied when the speed profiles for p_1 and p_2 are computed separately.

The following proposition (whose proof can be found in [17]) states the theoretical complexity of a simplified version of Problem III.1, called BASP-C, in which maximum and minimum acceleration and speed are constant on each arc.

Proposition III.3: Problem BASP-C is NP-hard.

IV. k-BASP

As we will see in Remark IV.6, SP can be viewed as a special case of the BASP, namely a BASP with unbounded acceleration limits. In fact, also BASP can be viewed as an SP but defined on a different graph with respect to the original one. More precisely, here, we introduce some restrictions on parameters \mathbb{B} that allow reducing the BASP to a standard SP that can be solved by Dijkstra's algorithm on an extended graph. Let $p \in P$, define

$$\ell^+(p) = \min\{\{\lambda \in [0, \ell(p)] \mid \int_0^\lambda \alpha_p^+(q) dq = \mu_p^+(\lambda)\}, +\infty\};$$

$$\ell^-(p) = \max\{\{\lambda \in [0, \ell(p)] \mid -\int_\lambda^{\ell(p)} \alpha_p^-(q) dq = \mu_p^-(\lambda)\}, -\infty\}.$$

In this way, $\ell^+(p)$ is the smallest value of $\lambda \in [0, \ell(p)]$ for which the solution of F in (2), with $\alpha^+ = \alpha_p^+$, starting from initial condition $F(0) = 0$, reaches the squared speed upper bound $\mu^+(\lambda)$. Note that $\ell^+(p) = \infty$ if no such value of λ exists. Similarly, $\ell^-(p)$ is the largest value of $\lambda \in [0, \ell(p)]$ for which the solution of B in (3), with $\alpha^- = \alpha_p^-$, starting from initial condition $B(\ell(p)) = 0$, reaches $\mu^+(\lambda)$. Again, $\ell^-(p) = -\infty$ if no such value of λ exists. Note that if $p, t, pt \in P$, $\ell^+(pt) \leq \ell^+(p)$ and $\ell^-(pt) \geq \ell^-(p)$ (actually, equalities hold if the values are all finite). Finally, we define

$$K(\mathbb{B}) = \min\{k \in \mathbb{N} \mid (\forall p \in P_s) |p| \geq k \Rightarrow \ell^+(p) \leq \ell^-(p)\}. \quad (4)$$

We call k -BASP any instance of Problem III.1 that satisfies $K(\mathbb{B}) \leq k$. For instance, consider the following chain graph $\mathbb{G} = (\mathbb{V} = \{s, 1, 2, f\}, \mathbb{E} = \{(s, 1), (1, 2), (2, f)\})$. Here, $(\forall \theta \in \mathbb{E}) \alpha^-(\theta) = -1$, $\alpha^+(\theta) = 1$, $\mu^-(\theta) = 0$, $\ell(\theta) = 1$, and $\mu^+((s, 1)) = 1$, $\mu^+((1, 2)) = \frac{2}{3}$, $\mu^+((2, f)) = 1$. In this case, $P_s = \{s, s1, s12, s12f\}$. Moreover, $K(\mathbb{B}) > 2$, since $\ell^+(s1) = 1 > 0 = \ell^-(s1)$, as reported in Fig. 2. Furthermore, $\ell^+(s12) < \ell^-(s12)$ and $\ell^+(12f) < \ell^-(12f)$ and $s12, 12f$ are the only paths of length 3. Fig. 3 shows the computation of $\ell^+(s12)$ and $\ell^-(s12)$; the computation of $\ell^+(12f)$ and $\ell^-(12f)$ is analogous. Hence, in this example, $K(\mathbb{B}) = 3$.

Note that $K(\mathbb{B}) - 1$ represents the maximum number of nodes of a path that can be traveled with a speed profile of maximum acceleration, followed by one of maximum deceleration, starting and ending with null speed, without violating the maximum speed constraint. The following

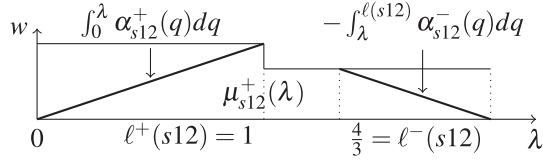


Fig. 3. Computation of $\ell^+(s12) = 1$ and $\ell^-(s12) = \frac{4}{3}$.

273 expression provides a simple upper bound on $K(\mathbb{B})$:

$$K(\mathbb{B}) \leq 1 + \left[2 \max_{\theta \in \mathbb{E}} \frac{\max_{\lambda \in [0, \ell(\theta)]} \mu^+(\theta, \lambda)}{\min_{\lambda \in [0, \ell(\theta)]} (\alpha^+(\theta, \lambda) \wedge |\alpha^-(\theta, \lambda)|) \ell(\theta)} \right]. \quad (5)$$

274 Note that $K(\mathbb{B}) = 1$ only if $\alpha_- = -\infty$ and $\alpha^+ = +\infty$, that is, if we
 275 do not consider acceleration bounds. We will present an algorithm that
 276 solves the k -BASP in polynomial time complexity with respect to $|\mathbb{V}|$
 277 and $|\mathbb{E}|$. However, note that the complexity is exponential with respect
 278 to k so that a correct estimation of $K(\mathbb{B})$ is critical. In general, the
 279 bound (5) overestimates $K(\mathbb{B})$. In Section V, we will present a simple
 280 method for correctly estimating $K(\mathbb{B})$.

281 We recall that \mathbb{V}_k represents the subset of language \mathbb{V}^* composed of
 282 strings with maximum length k , including the empty string ε . Define
 283 $\text{Suff}_k : P \rightarrow \mathbb{V}_k$ such that, if $|p| \leq k$, $\text{Suff}_k(p) = p$ and if $|p| > k$,
 284 $\text{Suff}_k(p)$ is the suffix of p of length k . The function Suff_k allows to
 285 introduce a partially defined transition function $\Gamma : \mathbb{V}_k \times \mathbb{V} \rightarrow \mathbb{V}_k$ by
 286 setting $\Gamma(r, \sigma) = \text{Suff}_k(r\sigma)$ if $r\sigma \in P$, otherwise, if $r\sigma \notin P$, $\Gamma(r, \sigma)$
 287 is not defined. Define the incremental cost function $\eta : P_s \times \mathbb{V} \rightarrow \mathbb{R}^+$
 288 such that, for $p \in P_s$ and $\sigma \in \mathbb{V}$, if $p\sigma \in P_s$, $\eta(p, \sigma) = T_{\mathbb{B}}(p\sigma) -$
 289 $T_{\mathbb{B}}(p)$, otherwise $\eta(p, \sigma) = +\infty$. In other words, $\eta(p, \sigma)$ is the dif-
 290 ference between the minimum time required for traversing $p\sigma$ and the
 291 minimum time required for traversing p . For simplicity of notation,
 292 from now on, we will denote $T_{\mathbb{B}}$ simply as T . The following proposition
 293 shows that the incremental cost is always strictly positive.

294 *Proposition IV.1:* $\eta(p, \sigma) \geq T(\sigma)$.

295 *Proof:* By 1) of Proposition III.2, $T(p\sigma) \geq T(p) + T(\sigma)$. \square

296 The following property, whose proof is presented in the Appendix,
 297 plays a key role in the solution algorithm.

298 *Proposition IV.2:* Let $p_1, p_2, t \in P$, if $p_1t, p_2t \in P$ and $\ell^+(t) \leq$
 299 $\ell^-(t)$, then $(\forall \sigma \in \mathbb{V}) T(p_1t\sigma) - T(p_1t) = T(p_2t\sigma) - T(p_2t)$.

300 The following is a direct consequence of Proposition IV.2. It states
 301 that, given $p \in P$ and $\sigma \in \mathbb{V}$, the incremental cost $\eta(p, \sigma)$ does not
 302 depend on the complete path p , but only on $\text{Suff}_k(p)$ (its last k symbols).

303 *Proposition IV.3:* If $K(\mathbb{B}) \leq k$ and $p, p' \in P$ are such that
 304 $\text{Suff}_k(p) = \text{Suff}_k(p')$, then $(\forall \sigma \in \mathbb{V}) \eta(p, \sigma) = \eta(p', \sigma)$.

305 Define function $\hat{\eta} : \mathbb{V}_k \times \mathbb{V} \rightarrow \mathbb{R}^+$, such that $\hat{\eta}(r, \sigma) = \eta(p, \sigma)$
 306 where $p \in P$ is any path such that $r = \text{Suff}_k(p)$. We set $\hat{\eta}(r, \sigma) = +\infty$
 307 if such path does not exist. Note that the function $\hat{\eta}$ is well-defined by
 308 Proposition IV.3, being $\eta(p, \sigma)$ identical among all paths p such that $r =$
 309 $\text{Suff}_k(p)$. In particular, Proposition IV.3 holds for $p' = \text{Suff}_k(p) = r$
 310 so that we can compute $\hat{\eta}$ as $\hat{\eta}(r, \sigma) = \eta(r, \sigma)$. In the following, since
 311 $\hat{\eta}$ is the restriction of η on $\mathbb{V}_k \times \mathbb{V}$, we denote $\hat{\eta}$ simply by η .

312 The value k can be viewed as the amount of memory required to
 313 solve the problem: once a node is reached, the optimal path from such
 314 node to the target one depends on the last k visited nodes. If $k = 1$, it
 315 only depends on the current node (i.e., no memory is required). This
 316 is the situation with the classical SP. More generally, $k > 1$ so that the
 317 optimal way to complete the path does not only depend on the current
 318 node, but also on the sequence of $k - 1$ nodes visited before reaching
 319 it. Define function $V : \mathbb{V}_k \rightarrow \mathbb{R}$ as

$$V(r) = \min_{p \in P_s | \text{Suff}_k p = r} T_{\mathbb{B}}(p). \quad (6)$$

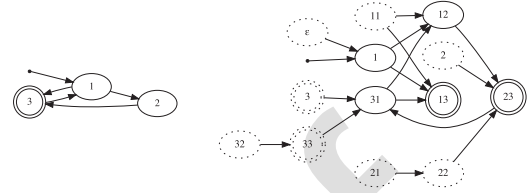


Fig. 4. Graph and its corresponding extension for $k = 2$.

Note that the solution of the BASP corresponds to $\min_{r \in \mathbb{V}_k | \tilde{r} \in F} V(r)$ 320
 (we recall that \tilde{r} is the last node of r). For $r \in \mathbb{V}_k$, define the set of 321
 predecessors of r as $\text{Prec}(r) = \{\tilde{r} \in \mathbb{V}_k \mid r = \Gamma(\tilde{r}, \tilde{r})\}$. The following 322
 proposition presents an expression for $V(r)$ that holds if $\ell^+(r') \leq$ 323
 $\ell^-(r')$ for all predecessors r' of r . 324

325 *Proposition IV.4:* Let $r \in \mathbb{V}_k$, if $(\forall r' \in \text{Prec}(r)) \ell^+(r') \leq \ell^-(r')$,
 326 then

$$V(r) = \min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r', \tilde{r})\}. \quad (7)$$

327 *Proof:* Let $S_r = \{q \in P_s \mid \text{Suff}_k q\tilde{r} = r\}$. $V(r) = \min_{p \in$ 328
 $P_s \mid \text{Suff}_k p = rT(p) = \min_{q \in S_r} \{T(q\tilde{r}) - T(q) + T(q)\} = \min_{q \in$ 329
 $S_r} \{T(q) + T((\text{Suff}_k q)\tilde{r}) - T(\text{Suff}_k q)\} = \min_{q \in S_r} \{T(q) +$ 330
 $\eta(\text{Suff}_k q, \tilde{r})\} = \min_{r' \in \text{Prec}(r), q \in S_{r'}} \{T(q) + \eta(r', \tilde{r})\} =$ 331
 $\min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r', \tilde{r})\}$, where we used the facts that 332
 $T(q\sigma) - T(q) = T(\text{Suff}_k q\sigma) - T(\text{Suff}_k q)$, by Proposition IV.2, 333
 and that $q \in P_s$ is such that $\text{Suff}_k q\tilde{r} = r \Leftrightarrow \text{Suff}_k q \in \text{Prec}(r)$. \blacksquare

334 As a consequence of Proposition IV.4, if $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$,
 335 $V(r)$ corresponds to the length of the shortest path from s to r on the
 336 extended directed graph $\tilde{\mathbb{G}} = (\tilde{\mathbb{V}}, \tilde{\mathbb{E}})$, where $\tilde{\mathbb{V}} = \mathbb{V}_k$ and $(r_1, r_2) \in \tilde{\mathbb{E}}$ 337
 if $r_2 = \Gamma(r_1, \tilde{r}_2)$ is defined, in this case its length is $\eta(r_1, \tilde{r}_2)$. The left 338
 part of Fig. 4 shows a graph consisting of three nodes. Node $s = 1$ is 339
 the source (indicated by the entering arrow) and the double border 340
 shows the final node $F = \{3\}$. The right part of Fig. 4 represents 341
 the corresponding extended graph, obtained for $k = 2$, consisting of 342
 13 nodes (the cardinality of \mathbb{V}_2). Note that some of the nodes are 343
 unreachable from the initial state, these are represented with dotted 344
 borders.

345 Solving k -BASP corresponds to finding a minimum-length path on
 346 $\tilde{\mathbb{G}}$ that connects node $s \in \mathbb{V}_k$ to $\tilde{F} = \{r \in \mathbb{V}_k \mid \tilde{r} \in F\}$. Note that the
 347 set of final states \tilde{F} for the extended graph $\tilde{\mathbb{G}}$ contains all paths $p \in \mathbb{V}_k$
 348 that end in an element of F . In the extended graph reported in Fig. 4, this
 349 corresponds to finding a minimum-length path from the starting node
 350 1 to one of the final nodes 3, 13, 23, and 33. Note that the unreachable
 351 nodes play no role in this procedure. We can find a minimum-length
 352 path by Dijkstra's algorithm applied to $\tilde{\mathbb{G}}$, leading to the following
 353 complexity result.

354 *Proposition IV.5:* k -BASP can be solved with complexity
 355 $O(|\mathbb{V}|^{k-1}|\mathbb{E}| + (|\mathbb{V}|^k \log |\mathbb{V}|^k))$.

356 *Proof:* Dijkstra's algorithm has time complexity $O(|E| +$ 357
 $|V| \log |V|)$, where $|E|$ and $|V|$ are the cardinalities of the edge
 358 and vertex sets, respectively. In our case, $|V| = |\tilde{\mathbb{V}}| = |\mathbb{V}_k| =$ 359
 $\sum_{i=0}^k |\mathbb{V}|^i = O(|\mathbb{V}|^k)$, $|E| = |\tilde{\mathbb{E}}| \leq |\mathbb{V}_{k-1}|\mathbb{E}| = O(|\mathbb{V}|^{k-1}|\mathbb{E}|)$. \square

360 The following remark establishes that SP can be viewed as a special
 361 case of the BASP without acceleration bounds.

362 *Remark IV.6:* If $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) \alpha^-(\theta, \lambda) = -\infty,$
 363 $\alpha^+(\theta, \lambda) = +\infty$, then $K(\mathbb{B}) = 1$. The resulting 1-BASP reduces to
 364 a standard SP on the graph \mathbb{G} and can be solved with time complexity
 365 $O(|\mathbb{E}| + |\mathbb{V}| \log |\mathbb{V}|)$.

V. ADAPTIVE A* ALGORITHM FOR k -BASP

366 The computation method based on Dijkstra's algorithm on the
 367 extended graph $\tilde{\mathbb{G}}$, presented in the previous section, has two main
 368

disadvantages. First, $\tilde{\mathbb{G}}$ has $\sum_{j=0}^k |\mathbb{V}|^j$ nodes so that the time required by Dijkstra's algorithm grows exponentially with k . We will show that it is possible to mitigate this problem and reduce the number of visited nodes by using the A* algorithm with a suitable heuristic. Second, the estimation of $k = K(\mathbb{B})$ from its definition is not an easy task. We will show that it is quite easy to adaptively find the correct value of k by starting from $k = 2$ and increasing k if needed.

The A* algorithm is a heuristic method that allows to compute the optimal path, if it exists (see [18]), by exploring the graph beginning from the starting node along the most promising directions according to a heuristic function that estimates the cost from the current position to the target node. Hence, to implement the A* algorithm, we need to define a heuristic function $h : \mathbb{V}_k \rightarrow \mathbb{R}$, such that, for $r \in \mathbb{V}_k$, $h(r)$ is a lower bound on $\min_{p \in P_{\tilde{r}, \tilde{F}}} T(p)$, that is, the minimum time needed for traveling from \tilde{r} to a final state in \tilde{F} . In general, we can compute lower bounds for the BASP by relaxing the acceleration constraints α^- and α^+ . Namely, let $\tilde{\mathbb{B}}$ be a parameter set obtained by relaxing acceleration constraints in \mathbb{B} . Then, if $K(\tilde{\mathbb{B}}) < K(\mathbb{B})$, by Proposition IV.5, the solution of the BASP for parameter $\tilde{\mathbb{B}}$ can be computed with a lower computational time than the solution with parameter \mathbb{B} . In particular, we obtain a very simple lower bound by removing acceleration bounds altogether, that is, by setting $\alpha^- = -\infty$ and $\alpha^+ = +\infty$. In this way, the vehicle is allowed to travel at maximum speed everywhere along the path and the incremental cost function $\eta(p, \sigma)$ is given by the time needed to travel γ_σ at maximum speed, that is, $\eta(p, \sigma) = \int_0^{\ell(\tilde{p}\sigma)} \frac{1}{\sqrt{\mu^+((\tilde{p}, \sigma), \lambda)}} d\lambda$.

Define the heuristic $h : \mathbb{V}_k \rightarrow \mathbb{R}^+$ as

$$h(r) = \min_{p \in P_{\tilde{r}, \tilde{F}}} T_{\tilde{\mathbb{B}}}(p). \quad (8)$$

Note that, if $\alpha^- = -\infty$ and $\alpha^+ = +\infty$, h corresponds to the solution of 1-BASP and all values of h can be efficiently precomputed by Dijkstra's algorithm (see Remark IV.6). The following proposition shows that h is admissible and consistent so that the A* algorithm, with heuristic h , provides the optimal solution of the k -BASP and its time complexity is no worse than Dijkstra's algorithm (see [19, Th. 2.9 and 2.10]).

Proposition V.1: Heuristic h satisfies the following two properties.

- 1) Admissibility: $(\forall r \in \mathbb{V}_k) h(r) \leq \min_{q \in P_{\tilde{r}, \tilde{F}}} T_{\tilde{\mathbb{B}}}(q)$.
- 2) Consistency: $(\forall r \in \mathbb{V}_k) (\forall \sigma \in \mathbb{V}) h(r) \leq \eta(r, \sigma) + h(\Gamma(r, \sigma))$.

Proof: 1) $h(r) = \min_{p \in P_{\tilde{r}, \tilde{F}}} T_{\tilde{\mathbb{B}}}(p) \leq \min_{q \in P_{\tilde{r}, \tilde{F}}} T_{\tilde{\mathbb{B}}}(q)$, since $\tilde{\mathbb{B}}$ is a relaxation of \mathbb{B} .

2) $h(r) = \min_{p \in P_{\tilde{r}, \tilde{F}}} T_{\tilde{\mathbb{B}}}(p) \leq T_{\tilde{\mathbb{B}}}(\sigma) + \min_{p \in P_{\sigma, \tilde{F}}} T_{\tilde{\mathbb{B}}}(p) \leq T_{\tilde{\mathbb{B}}}(\sigma) + \min_{p \in P_{\sigma, \tilde{F}}} T_{\mathbb{B}}(p) \leq \eta(r, \sigma) + h(\Gamma(r, \sigma))$, where $T_{\tilde{\mathbb{B}}}(\sigma) \leq T_{\mathbb{B}}(\sigma)$ by 2) of Proposition III.2 and $T_{\tilde{\mathbb{B}}}(p) \leq T_{\mathbb{B}}(p)$ by Proposition IV.1. \square

Since heuristic h is admissible and consistent, A* is equivalent to Dijkstra's algorithm, with the only difference that the incremental cost function $\eta(r, \sigma)$ is replaced by the modified cost

$$\tilde{\eta}(r, \sigma) = \eta(r, \sigma) + h(\Gamma(r, \sigma)) - h(r) \quad (9)$$

(see [19, Lemma 2.3] for a complete discussion). A description of the A* algorithm can be found in literature (for instance, see [19, Algorithm 2.13]). We define a priority queue \mathcal{Q} that contains open nodes, that is, nodes that have already been generated but have not yet been visited. Namely, \mathcal{Q} is an ordered set of pairs $(r, t) \in \mathbb{V}_k \times \mathbb{R}^+$, in which $r \in \mathbb{V}_k$ and t is a lower bound for the time associated to the best completion of r to a path arriving at a final state. We need to perform the following operations on \mathcal{Q} : operation $\text{Insert}(\mathcal{Q}, (r, t))$ inserts couple (r, t) into \mathcal{Q} ; operation $(r, t) = \text{DeleteMin}(\mathcal{Q})$ returns the first couple of \mathcal{Q} , that is, the couple (r, t) with the minimum time t , and removes this couple from \mathcal{Q} ; and, operation $\text{DecreaseKey}(\mathcal{Q}, (r, t))$ assumes that \mathcal{Q} already contains a couple (r, t') with $t' > t$ and substitutes this

couple with (r, t) . Furthermore, we consider three partially defined maps $\text{value} : \mathbb{V}_k \rightarrow \mathbb{R}$, $\text{parent} : \mathbb{V}_k \rightarrow \mathbb{V}_k$, $\text{closed} : \mathbb{V}_k \rightarrow \{0, 1\}$, such that, for $r \in \mathbb{V}_k$, $\text{value}(r)$ is the current best upper estimate of $V(r)$, $\text{parent}(r)$ is the parent node of r , and $\text{closed}(r) = 1$ if node r has already been visited. Maps value , parent , and closed can be implemented as hash tables.

Algorithm V.2 (A algorithm for k -BASP):*

1) [initialization] Set $\mathcal{Q} = \{(s, h(s))\}$, $\text{value}(s) = 0$.

2) [expansion] Set $(r, t) = \text{DeleteMin}(\mathcal{Q})$ and set $\text{closed}(r) = 1$. If $\tilde{r} \in \tilde{F}$, then t is the optimal solution and the algorithm terminates, returning maps value , parent . Otherwise, for each $\sigma \in \mathbb{V}$ for which $\Gamma(r, \sigma)$ is defined, set $r' = \Gamma(r, \sigma)$, $t' = t + \tilde{\eta}(r, \sigma)$. If $\text{closed}(r') = 1$, go to 3). Else, if $\text{value}(r')$ is undefined $\text{Insert}(\mathcal{Q}, (r', t'))$. Otherwise, if $t' < \text{value}(r')$, set $\text{value}(r') = t'$, $\text{parent}(r') = r$ and do $\text{DecreaseKey}(\mathcal{Q}, (r', t'))$.

3) [loop] If $\mathcal{Q} \neq \emptyset$ go back to 2), otherwise no solution exists.

Proposition V.3: Algorithm V.2 terminates and returns the optimal solution (if it exists), with a time-complexity not higher than Dijkstra's algorithm on the extended graph $\tilde{\mathbb{G}}$.

Proof: It is a consequence of the fact that heuristic h is admissible and consistent (see [19, Th. 2.9 and 2.10]). \square

Note that, at the end of Algorithm V.2, $\text{value}(f)$ is the optimal value of the k -BASP and the optimal path from s to set F can be reconstructed from map parent .

One possible limitation of Algorithm V.2 is that estimating $K(\mathbb{B})$ from its definition can be difficult. A correct estimation of $K(\mathbb{B})$ is critical for the efficiency of the algorithm. Indeed, if $K(\mathbb{B})$ is overestimated, the time complexity of the algorithm is higher than it would be with a correct estimate. On the other hand, if $K(\mathbb{B})$ is underestimated, Algorithm V.2 is not correct since Proposition IV.4 does not hold. Here, we propose an algorithm that adaptively finds a suitable value for k in Algorithm V.2, such that $k \leq K(\mathbb{B})$, but, in any case, allows to find the optimal solution of the BASP. First, we define the modified cost function $W : \mathbb{V}_k \rightarrow \mathbb{R}$ as $W(r) = V(r) + h(r)$, where V is given by (6) and h is the heuristic given by (8). If $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$, then W is the solution of

$$\begin{cases} W(r) = \min_{r' \in \text{Prec}(r)} \{W(r') + \tilde{\eta}(r, r')\} \\ W(s) = h(s). \end{cases} \quad (10)$$

Indeed, following the same steps of the proof of Proposition IV.4, $W(r) = V(r) + h(r) = \min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r, r') + h(r) + h(r') - h(r')\} = \min_{r' \in \text{Prec}(r)} \{W(r') + \tilde{\eta}(r, r')\}$. Hence, $W(r)$ corresponds to the length of the shortest path from s to r on $\tilde{\mathbb{G}}$, with arc length given according to $\tilde{\eta}$. If condition $\ell^+(r) \leq \ell^-(r)$ is not satisfied for all $r \in \mathbb{V}_k$, (10) does not hold for all $r \in \mathbb{V}_k$ and W does not represent the solution of an SP. However, the following proposition shows that we can still find a lower bound \hat{W} of W that does correspond to the solution of an SP.

Proposition V.4: Let $\hat{W} : \mathbb{V}_k \rightarrow \mathbb{R}$ be the solution of

$$\begin{cases} \hat{W}(r) = \min_{r' \in \text{Prec}(r)} \{\hat{W}(r') + \hat{\eta}(r, r')\} \\ \hat{W}(s) = 0, \end{cases} \quad (11)$$

where if $\ell^+(r') \leq \ell^-(r')$ or $|r'| < k$, $\hat{\eta}(r, r') = \tilde{\eta}(r, r')$, otherwise $\hat{\eta}(r, r') = h(r) - h(r')$. Then, $(\forall r \in \mathbb{V}_k)$

1) $\hat{W}(r) \leq W(r)$;

2) $(\forall \tilde{r} \in \mathbb{V}_k \mid \tilde{W}(\tilde{r}) \leq \hat{W}(r)) \ell^+(\tilde{r}) \leq \ell^-(\tilde{r}) \Rightarrow \hat{W}(r) = W(r)$.

Proof: 1) For $r \in \mathbb{V}_k$, let $p \in P_s$ be such that $\text{Suff}_k p \in \text{Prec}(r)$. If $\ell^+(\text{Suff}_k p) \leq \ell^-(\text{Suff}_k p)$, in view of Proposition IV.2, $T(p\tilde{r}) = T(p) + \eta(\text{Suff}_k p, \tilde{r})$, otherwise, obviously, $T(p\tilde{r}) \geq T(p)$. Hence, in both cases, by the definition of $\tilde{\eta}$ in (9), $T(p\tilde{r}) + h(r) \geq T(p) + h(\text{Suff}_k p) + \tilde{\eta}(\text{Suff}_k p, \tilde{r})$. By contradiction, assume

480 $(\exists A \subset \mathbb{V}_k) A \neq \emptyset$ such that $(\forall r \in A) \hat{W}(r) > W(r)$. Let
 481 $\bar{r} = \operatorname{argmin}_{\bar{r} \in A} W(\bar{r})$ and $S_{\bar{r}} = \{q \in P_s \mid \operatorname{Suff}_k q \in \operatorname{Prec}(\bar{r})\}$,
 482 then $W(\bar{r}) = V(\bar{r}) + h(\bar{r}) = \min_{p \in P_s \mid \operatorname{Suff}_k p = \bar{r}} T(p) + h(\bar{r}) =$
 483 $\min_{q \in S_{\bar{r}}} T(q\bar{r}) + h(\bar{r}) \geq \min_{q \in S_{\bar{r}}} \{T(q) + h(\operatorname{Suff}_k(q)) + \hat{\eta}(\operatorname{Suff}_k q,$
 484 $\bar{r})\} = \min_{r' \in \operatorname{Prec}(\bar{r})} \{\hat{W}(r') + \hat{\eta}(r', \bar{r})\} = \hat{W}(\bar{r})$, where we used the
 485 fact that $W(r') = \hat{W}(r')$, that follows from the definition of \bar{r} , since
 486 the value of r' that attains the minimum is such that $W(r') < W(\bar{r})$.
 487 Then, the obtained inequality contradicts the fact that $\hat{W}(\bar{r}) > W(\bar{r})$.
 488 2) Let $A \subset \mathbb{V}$ be the set of values of $r \in \mathbb{V}$ for which 2)
 489 does not hold, and by contradiction, assume that $A \neq \emptyset$ and let
 490 $\hat{r} = \operatorname{argmin}_{r \in A} \hat{W}(r)$. Then, by definition of \hat{r} , it satisfies the
 491 following two properties: $(\forall \bar{r} \in \mathbb{V}_k \mid \hat{W}(\bar{r}) \leq \hat{W}(\hat{r})) \ell^+(\bar{r}) \leq \ell^-(\bar{r})$,
 492 moreover, $\hat{W}(\hat{r}) \neq W(\hat{r})$. Note that, from the definitions of \hat{W} ,
 493 $W(s) = \hat{W}(s)$. Then, $W(\hat{r}) = \min_{p \in P_s \mid \operatorname{Suff}_k p = \hat{r}} T(p) + h(\hat{r}) =$
 494 $\min_{q \in P_s \mid \operatorname{Suff}_k q \in \operatorname{Prec}(\hat{r})} \{T(q\hat{r}) + h(\operatorname{Suff}_k q) - h(\operatorname{Suff}_k q) + h(\hat{r})\} =$
 495 $\min_{r' \in \operatorname{Prec}(\hat{r})} \{\hat{W}(r') + \hat{\eta}(r', \hat{r})\} = \hat{W}(\hat{r})$, which contradicts the
 496 definition of \hat{r} . Here, we used (9) and the fact that, since $\hat{W}(r') < \hat{W}(\hat{r})$
 497 and by the definition of \hat{r} , $\hat{W}(r') = W(r')$. \square

498 Proposition V.4 implies that $\hat{W}(r)$ is a lower bound of $W(r)$ and
 499 that it corresponds to the length of the shortest path from s to r on
 500 the extended directed graph \mathbb{G} , with arc length given in accordance
 501 to (11), namely by the value of function $\hat{\eta}$. Hence, $\hat{W}(f)$ can be
 502 computed by Dijkstra's algorithm (which is equivalent to compute V
 503 with A^* algorithm, with heuristic h). The algorithm that we are going
 504 to present is based on the following basic observation. If A^* algorithm
 505 computes $f^* = \operatorname{argmin}_{f \in \bar{F}} \hat{W}(f)$ by visiting only nodes for which
 506 $\ell^+(r) \leq \ell^-(r)$, then 2) of Proposition V.4 is satisfied for $r = f^*$ and
 507 $\hat{W}(f^*) = W(f^*)$ is the optimal value of the k -BASP. If this is not the
 508 case, we increase k by 1 and rerun the A^* algorithm. Note that the
 509 algorithm starts with $k = 2$, since, according to its definition, $K(\mathbb{B})$
 510 equals 1 only if no acceleration bounds are present and, in this case, the
 511 BASP is equivalent to a standard SP and can be solved by Dijkstra's
 512 algorithm.

513 *Algorithm V.5 (Adaptive A^* algorithm for k -BASP):*

514 1) Set $k = 2$.

515 2) Execute A^* algorithm, and at every visit of a new node r , if none
 516 of the two conditions $\ell^+(r) \leq \ell^-(r)$ and $|r| < k$ holds, set $k = k + 1$
 517 and repeat step 2).

518 Note that the algorithm does not compute the exact value $K(\mathbb{B})$.
 519 Rather, it underestimates it. More precisely, it stops with the smallest
 520 k value needed to solve the BASP between the given source and
 521 destination nodes. That is, the smallest k that satisfies the k -BASP
 522 definition over the explored subgraph.

523 *Proposition V.6:* Algorithm V.5 terminates with $k \leq K(\mathbb{B})$ and
 524 returns an optimal solution.

525 *Proof:* By Definition (4) of $K(\mathbb{B})$, if $k = K(\mathbb{B})$, the condition
 526 $\ell^+(r) \leq \ell^-(r)$ is satisfied for all r . Hence, there exists $k \leq K(\mathbb{B})$
 527 for which the algorithm terminates. Let $r \in \mathbb{V}_k$, with $\bar{r} \in F$ be the
 528 last-visited node before the termination of the algorithm. By 2) of
 529 Proposition V.4, we have that $\hat{W}(r) = W(r) = V(r)$ (since $h(r) =$
 530 0), but, by definition, $V(r)$ is the shortest time for reaching a node in
 531 F . \square

532 VI. NUMERICAL EXPERIMENTS

533 A. Randomly Generated Problems

534 We performed various tests on problems associated to graphs with n
 535 nodes, for increasing values of n , randomly generated with function `ge-`
 536 `ographical_threshold_graph` of Python package `NetworkX` (`networkx.`
 537 `org`). Essentially, each node is associated to a position randomly chosen
 538 from set $[0, 1]^2$. Edges are randomly determined in such a way that

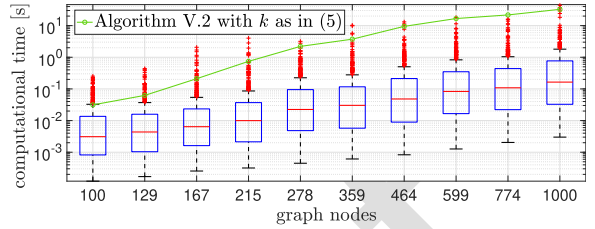


Fig. 5. BASP computing times on graphs of different size.

TABLE I
PERCENTAGES OF k VALUES FOR GRAPHS OF DIFFERENT SIZE

n	$k=3$	$k=4$	$k=5$	$k=6$	\bar{k}	n	$k=3$	$k=4$	$k=5$	$k=6$	\bar{k}
100	80.4%	18.0%	1.6%	0.0%	86	359	61.6%	33.8%	4.4%	0.2%	161
129	81.0%	17.2%	1.8%	0.0%	89	464	60.8%	33.0%	6.0%	0.2%	202
167	77.8%	19.6%	2.0%	0.6%	170	599	51.6%	39.8%	8.2%	0.4%	188
215	72.6%	24.2%	3.2%	0.0%	177	744	49.4%	43.0%	6.4%	1.2%	338
278	63.2%	30.6%	6.2%	0.0%	146	1000	43.6%	46.0%	9.6%	0.8%	300

539 closer nodes have a higher connection probability. We multiplied the
 540 obtained positions by factor $10\sqrt{n}$, in order to obtain the same average
 541 node density independently of n . Then, we associated a random angle θ_i
 542 to each node, obtained from a uniform distribution in $[0, 2\pi]$. In this way,
 543 each node of the random graph is associated to a vehicle configuration,
 544 consisting of a position and an angle. Set $\tau(\theta_i) = [\cos \theta_i, \sin \theta_i]^T$.
 545 Each edge (i, j) is associated to a *Dubins path*, which is defined as the
 546 shortest curve of bounded curvature that connects the configurations
 547 associated to nodes i and j , with initial tangent parallel to $\tau(\theta_i)$ and
 548 final tangent parallel to $\tau(\theta_j)$. We chose the minimum turning radius for
 549 the path associated to edge (i, j) as $r_{i,j} = \min\{\ell((i, j)) / (d(\theta_i, \theta_j)), 2\}$
 550 where $d(x, y)$ is the angular distance between angles x and y . We set
 551 the acceleration and deceleration bounds constant for all paths and
 552 equal to 0.1 ms^{-2} . The upper squared speed bound is constant for
 553 each arc and given by $2r$, where r is the minimum curvature radius
 554 of the path associated to the arc. In our tests, we used the adaptive
 555 A^* algorithm (see Algorithm V.5). First, we ran simulations for ten
 556 values of n , logarithmically spaced between 100 and 1000. For each
 557 n , we generated 50 different graphs, and for each one of them, we
 558 ran ten simulations, randomly choosing source and target nodes. Fig. 5
 559 shows the mean values and the distributions of the computational times
 560 of Algorithm V.5 and it also shows the mean computational times of
 561 Algorithm V.2 with k computed as in (5). Note that the mean times of
 562 Algorithm V.2 are at least one order of magnitude higher than those of
 563 Algorithm V.5. Table I shows, for each n , the percentages of k values
 564 returned by Algorithm V.5, and the mean value \bar{k} of k computed as
 565 in (5). Note that the values obtained with (5) are on average 54.8 times
 566 larger than those returned by Algorithm V.5.

567 In Section V, we showed that, for a given problem instance, path p^* ,
 568 corresponding to the solution of the BASP, is in general different from
 569 the path \hat{p} obtained as the solution of the BASP with infinite acceleration
 570 bounds (which, in fact, is an SP) and from the path \tilde{p} obtained as the
 571 solution of SP with edge costs equal to their lengths. We ran some
 572 numerical experiments to compare travel times $T_{\mathbb{B}}(p^*)$ and $T_{\mathbb{B}}(\hat{p})$,
 573 (i.e., the travel time of p^* and the one of \hat{p} on which speed has been
 574 planned using the same acceleration bounds of the BASP), and lengths
 575 $\ell(p^*)$ and $\ell(\tilde{p})$. Namely, we generated 50 different random graphs with
 576 $n = 100$ with the procedure presented previously. For each instance,
 577 we considered ten problems obtained by randomly choosing source and
 578 target nodes. Then, we solved the BASP with different acceleration
 579 bounds α^+ and α^- logarithmically spaced in $[0.01, 1] \text{ ms}^{-2}$, with
 580

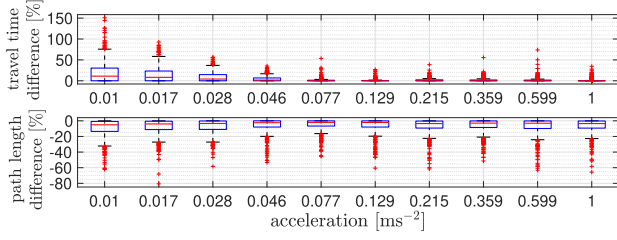


Fig. 6. Travel time difference between BASP and BASP without acceleration bounds and path length difference between BASP and SP with edge costs equal to their lengths.

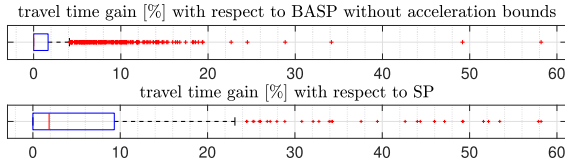


Fig. 7. Travel time gain of BASP on 1000 simulations on the 2 485-node graph with respect to the BASP without acceleration bounds and SP with edge costs equal to their lengths.

580 $\alpha^+ = \alpha^-$. In Fig. 6 (top), we compare the optimal travel times along paths p^* and \hat{p} , that is, for each value of the acceleration and deceleration bounds, we report the relative percentage difference $100 \frac{T_{\mathbb{B}}(\hat{p}) - T_{\mathbb{B}}(p^*)}{T_{\mathbb{B}}(p^*)}$ obtained for each test. We observe that for low acceleration and deceleration bounds the difference is more significant, while as the acceleration and deceleration bounds increase, the travel time difference between the two paths tends to be smaller. This is due to the fact that, if acceleration bounds are sufficiently high, paths p^* and \hat{p} are the same. In Fig. 6 (bottom), we compare the length of paths p^* and \tilde{p} showing how the BASP solution tends to differ from the SP with edge costs equal to their lengths even for small acceleration bounds. For p^* and \tilde{p} to coincide one needs even smaller acceleration bounds.

592 B. Real Industrial Applications

593 Here, we present a problem from a real industrial application representing an automated warehouse provided by packaging company
594 Ocme S.r.l., based in Parma, Italy. The problem is described by a graph of 2 485 nodes and 4 411 arcs. The acceleration and deceleration bounds are constant, equal for all arcs, and given by $\alpha^+ = 0.28 \text{ ms}^{-2}$ and $\alpha^- = -0.18 \text{ ms}^{-2}$. The speed bounds are constant for each arc but vary among different arcs, according to the associated path curvatures, and they take values on interval $[0.1, 1.7] \text{ ms}^{-1}$. The arc lengths take values in $[0.2, 18] \text{ m}$ and have an average value of 4.2 m. We ran 1000 simulations by randomly choosing source and the target nodes. The average value and the standard deviation of the computational time are 0.1587 and 1.9355 s, respectively. The mean value of k returned by Algorithm V.5 is 5, while the bound obtained with (5) is 105. We compare travel times $T_{\mathbb{B}}(p^*)$, $T_{\mathbb{B}}(\hat{p})$, and $T_{\mathbb{B}}(\tilde{p})$, that is, the travel time of p^* and the ones of \hat{p} and \tilde{p} on which speed has been planned using the same acceleration bounds of the BASP. Fig. 7 compares the optimal travel time gain obtained using p^* over \hat{p} and \tilde{p} . Namely, we report the relative percentage differences over 1000 tests. In the first case, we had a 2.17% mean gain and the 25% best performing paths p^* had a 8.53% mean gain over \hat{p} . While, in the latter case, we had a 5.85% mean gain and the 25% best performing paths p^* had a 14.16% mean gain over \tilde{p} . Note that these results are probably due to the fact that

the graph associated to the industrial problem has a low connectivity. Indeed, most nodes in the industrial problem represent positions in corridors and are connected only to the node preceding them and the one following them along the corridor. Nonetheless, in such industrial context, even moderate improvements represent a significant gain for a company.

APPENDIX

Proposition A.1: Let $\mu, \alpha : [0, +\infty) \rightarrow \mathbb{R}^+$, for $i \in \{1, 2\}$, let F_i be the solution of the differential equation (2) where F_i replaces F and $w_{0,i}$ replaces $\mu(0)$, with $0 \leq w_{0,i} \leq \mu(0)$; and let $\bar{\lambda}$ be such that $\mu(\bar{\lambda}) = \int_0^{\bar{\lambda}} \alpha(\lambda) d\lambda$. Then, $(\forall \lambda \geq \bar{\lambda}) F_1(\lambda) = F_2(\lambda)$.

Proof: Without loss of generality, assume that $w_{0,1} \geq w_{0,2}$. This implies that $(\forall \lambda \geq 0) F_1(\lambda) \geq F_2(\lambda)$. Indeed, assume by contradiction that there exists $\bar{\lambda}$ such that $F_1(\bar{\lambda}) < F_2(\bar{\lambda})$, then, by continuity of F_1 and F_2 , this implies that there exists $\hat{\lambda} \leq \bar{\lambda}$ such that $F_1(\hat{\lambda}) = F_2(\hat{\lambda})$, thus $(\forall \lambda \geq \hat{\lambda}) F_1(\lambda) = F_2(\lambda)$, since, for $\lambda \geq \hat{\lambda}$, $F_1(\lambda)$ and $F_2(\lambda)$ solve the same differential equation with the same initial condition at $\lambda = \hat{\lambda}$, contradicting the assumption. Furthermore, note that $(\exists \bar{\lambda} \in (0, \bar{\lambda}]) F_2(\bar{\lambda}) = \mu(\bar{\lambda})$. Indeed, if by contradiction $(\forall \lambda \in (0, \bar{\lambda}]) F_2(\lambda) < \mu(\lambda)$, then $(\forall \lambda \in (0, \bar{\lambda}]) F_2'(\lambda) = \alpha(\lambda)$ so that $F_2(\bar{\lambda}) - F_2(0) = \int_0^{\bar{\lambda}} \alpha(\lambda) d\lambda = \mu(\bar{\lambda})$, which contradicts the assumption. Hence, $(\exists \bar{\lambda} \in \mathbb{R}^+) F_2(\bar{\lambda}) = F_1(\bar{\lambda}) = \mu(\bar{\lambda})$, and consequently, $(\forall \lambda \geq \bar{\lambda}) F_1(\lambda) = F_2(\lambda)$, which implies the thesis, being $\bar{\lambda} \geq \hat{\lambda}$. \square

For $p \in P, \lambda \in [0, \ell(p)]$, we set $\mathcal{W}_p(\lambda) = w$, where w is the solution of Problem (1) for path p . In other words, $\mathcal{W}_p(\lambda)$ is the square of the optimal speed profile for traversing the path p , evaluated at arc length λ , with respect to p .

Proposition A.2 1): Let $p_1, p_2, q \in P$, be such that $p_1 q, p_2 q \in P$, then $(\forall \lambda \geq \ell^+(q)) \mathcal{W}_{p_1 q}(\ell(p_1) + \lambda) = \mathcal{W}_{p_2 q}(\ell(p_2) + \lambda)$.

2) Let $p, q_2, q_1 \in P$, be such that $p q_1, p q_2 \in P$, then $(\forall \lambda \leq \ell^-(p)) \mathcal{W}_{p q_1}(\lambda) = \mathcal{W}_{p q_2}(\lambda)$.

Proof: We only prove 1), the proof of 2) is analogous. Note that, for $\lambda \geq 0$, $\mathcal{W}_{p_1 q}(\lambda + \ell(p_1)) = \min\{F_1(\lambda), B(\lambda)\}$, $\mathcal{W}_{p_2 q}(\lambda + \ell(p_2)) = \min\{F_2(\lambda), B(\lambda)\}$, where F_1 and F_2 are the solution of (2) with $\mu = \mu^+$ and initial conditions $w_{0,1} = \mathcal{W}_{p_1}(\ell(p_1))$ and $w_{0,2} = \mathcal{W}_{p_2}(\ell(p_2))$, respectively, and B is the solution of (3) with $\mu = \mu^+$. By Proposition A.1, for $\lambda \geq \ell^+(q)$, $F_1(\lambda) = F_2(\lambda)$. Consequently, $(\forall \lambda \geq \ell^+(q)) \mathcal{W}_{p_1 q}(\ell(p_1) + \lambda) = \mathcal{W}_{p_2 q}(\ell(p_2) + \lambda)$. \square

A. Proof of Proposition IV.2

Let Ψ be defined as in (1a), then $T(p_1 t \sigma) - T(p_1 t) = \int_0^{\ell(p_1 t \sigma)} \Psi(\mathcal{W}_{p_1 t \sigma}(\lambda)) d\lambda - \int_0^{\ell(p_1 t)} \Psi(\mathcal{W}_{p_1 t}(\lambda)) d\lambda = \int_{\ell(p_1) + \ell^-(t)}^{\ell(p_1 t \sigma)} \Psi(\mathcal{W}_{p_1 t \sigma}(\lambda)) d\lambda - \int_{\ell(p_1) + \ell^-(t)}^{\ell(p_1 t)} \Psi(\mathcal{W}_{p_1 t}(\lambda)) d\lambda$, where we used that, by 2) of Proposition A.2, $(\forall \lambda \leq \ell(p_1) + \ell^-(t)) \Psi(\mathcal{W}_{p_1 t \sigma}(\lambda)) = \Psi(\mathcal{W}_{p_1 t}(\lambda))$. Similarly, $T(p_2 t \sigma) - T(p_2 t) = \int_{\ell(p_2) + \ell^-(t)}^{\ell(p_2 t \sigma)} \Psi(\mathcal{W}_{p_2 t \sigma}(\lambda)) d\lambda - \int_{\ell(p_2) + \ell^-(t)}^{\ell(p_2 t)} \Psi(\mathcal{W}_{p_2 t}(\lambda)) d\lambda$. Moreover, by 1) of Proposition A.2, we have that $(\forall \lambda \geq \ell^+(t \sigma)) \mathcal{W}_{p_1 t \sigma}(\ell(p_1) + \lambda) d\lambda = \mathcal{W}_{p_2 t \sigma}(\ell(p_2) + \lambda) d\lambda$ and $(\forall \lambda \geq \ell^+(t)) \mathcal{W}_{p_1 t}(\ell(p_1) + \lambda) d\lambda = \mathcal{W}_{p_2 t}(\ell(p_2) + \lambda) d\lambda$, which imply that $T(p_1 t \sigma) - T(p_1 t) = T(p_2 t \sigma) - T(p_2 t)$, since $\ell^+(t) \leq \ell^-(t)$, and as noticed in Section IV, $\ell^+(t \sigma) \leq \ell^+(t)$. \blacksquare

REFERENCES

- [1] A. Abbadi and V. Prenosil, "Safe path planning using cell decomposition approximation," in *Proc. Int. Conf. Distance Learn. Simul. Commun.*, May 2015, pp. 8–14.
- [2] R. Seif and M. A. Oskoei, "Mobile robot path planning by RRT* in dynamic environments," *Int. J. Intell. Syst. Appl.*, vol. 7, no. 5, pp. 24–30, 2015.

- Q3
- 670 [3] R. V. Cowlagi and P. Tsiotras, "Shortest distance problems in graphs using
671 history-dependent transition costs with application to kinodynamic path
672 planning," in *Proc. Amer. Control Conf.*, 2009, pp. 414–419.
- 673 [4] B. C. Dean, "Shortest paths in FIFO time-dependent networks: Theory and
674 algorithms," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech.
675 Rep., 2004.
- 676 [5] E. He, N. Boland, G. Nehmauser, and M. Savelsbergh, "Time-dependent
677 shortest path problems with penalties and limits on waiting," *INFORMS*
678 *J. Comput.*, vol. 33, no. 3, pp. 997–1014, 2021.
- 679 [6] I. Ioachim, S. Gélinas, F. Soumis, and J. Desrosiers, "A dynamic program-
680 ming algorithm for the shortest path problem with time windows and linear
681 node costs," *Networks*, vol. 31, pp. 193–204, 1998.
- 682 [7] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. New
683 York, NY, USA: McGraw-Hill, 1971.
- 684 [8] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artif.*
685 *Intell.*, vol. 155, no. 1–2, pp. 93–146, 2004.
- 686 [9] D. Ferguson and A. Stentz, "Field D*: An interpolation-based path planner
687 and replanner," *Springer Tracts Adv. Robot.*, vol. 28, pp. 239–253, 2007.
- 688 [10] D. H. Kim, N. T. Hai, and W. Y. Joe, "A guide to selecting path planning
689 algorithm for automated guided vehicle (AGV)," in *AETA 2017—Recent*
690 *Advances in Electrical Engineering and Related Sciences: Theory and*
691 *Application*. Cham, Switzerland: Springer, 2018, pp. 587–596.
- 692 [11] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The
693 path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89,
694 1986.
- [12] M. Raineri, S. Perri, and C. G. L. Bianco, "Online velocity planner for
695 laser guided vehicles subject to safety constraints," in *Proc. IEEE/RSJ Int.*
696 *Conf. Intell. Robots Syst.*, Sep. 2017, pp. 6178–6184. 697
- [13] D. Lam, C. Manzie, M. C. Good, and R. Bitmead, "Receding horizon
698 time-optimal control for a class of differentially flat systems," *Syst. Control*
699 *Lett.*, vol. 83, pp. 61–66, 2015. 700
- [14] T. Faulwasser, V. Hagemeyer, and R. Findeisen, "Optimal exact path-
701 following for constrained differentially flat systems," in *Proc. 18th World*
702 *Congr. Int. Fed. Autom. Control*, 2011, pp. 9875–9880. 703
- [15] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M.
704 Diehl, "Time-optimal path tracking for robots: A convex optimization
705 approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327,
706 Oct. 2009. 707
- [16] L. Consolini, M. Laurini, M. Locatelli, and A. Minari, "A solution of
708 the minimum-time speed planning problem based on lattice theory," *J.*
709 *Franklin Inst.*, vol. 357, pp. 7617–7637, 2020. 710
- [17] S. Ardizzoni, L. Consolini, M. Laurini, and M. Locatelli, "The bounded
711 acceleration shortest path problem: Complexity and solution algo-
712 rithms," 2021, *arXiv:2103.02914*. 713 Q4
- [18] D. Gelperin, "On the optimality of A*," *Artif. Intell.*, vol. 8, no. 1,
714 pp. 69–76, 1977. 715
- [19] S. Edelkamp and S. Schroedl, *Heuristic Search: Theory and Applications*.
716 Elsevier Science, 2011. 717 Q5

GENERAL INSTRUCTION

- **Authors:** Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections.
- **Authors:** We cannot accept new source files as corrections for your article. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

QUERIES

- Q1. Author: Please supply index terms/keywords for this article. To download the IEEE Taxonomy go to https://www.ieee.org/documents/taxonomy_v101.pdf.
- Q2. Author: Please confirm or add details for any funding or financial support for the research of this article.
- Q3. Author: Please provide the report number in Ref. [4].
- Q4. Author: Please update Ref. [17]
- Q5. Author: Please provide the location of the publisher in Ref. [19].

Technical Notes and Correspondence

Solution Algorithms for the Bounded Acceleration Shortest Path Problem

Stefano Ardizzoni , Luca Consolini , Mattia Laurini , and Marco Locatelli 

Abstract—The purpose of this article is to introduce and characterize the bounded acceleration shortest path problem (BASP), a generalization of the shortest path problem (SP). This problem is associated to a graph: nodes represent positions of a mobile vehicle and arcs are associated to preassigned geometric paths that connect these positions. The BASP consists in finding the minimum-time path between two nodes. Differently from the SP, the vehicle has to satisfy bounds on maximum and minimum acceleration and speed, which depend on the vehicle's position on the currently traveled arc. Even if the BASP is NP-hard in the general case, we present a solution algorithm that achieves polynomial time-complexity under some additional hypotheses on problem data.

Index Terms—

I. INTRODUCTION

The combinatorial problem of detecting the best path from a source to a destination node over an oriented graph with *constant* costs associated to its arcs, also known as shortest path problem (SP in what follows), is well known and can be efficiently solved, e.g., by the Dijkstra algorithm (in case of nonnegative costs). The continuous problem of minimum-time speed planning over a *fixed* path under given speed and acceleration constraints, also depending on the position along the path, is also widely studied and very efficient algorithms for its solution exist. But the combination of these two problems, called in what follows bounded acceleration shortest path problem (BASP), turns out to be more challenging than the two problems considered separately. More precisely, in terms of the complexity theory, it is possible to prove that the BASP is NP-hard, while the two problems considered separately are both polynomially solvable. In the BASP, we still have the combinatorial search for a best path as in SP but, differently from SP, the cost of an arc (more precisely, the time to traverse it) is not a constant value but depends on the speed planning along the arc itself, which, in turn, depends on the speed and acceleration constraints not only over the same arc but also over those preceding and following it in the selected path. Fig. 1(a) presents a simple scenario that allows

Manuscript received October 25, 2021; accepted April 15, 2022. This work was supported by the Programme "FIL-Quota Incentivante" of University of Parma and co-sponsored by Fondazione Cariparma. Recommended by Associate Editor T. Faulwasser. (*Corresponding author: Luca Consolini.*)

The authors are with the Dipartimento di Ingegneria e Architettura, Università degli Studi di Parma, 43124 Parma, Italy (e-mail: stefano.ardizzoni@unipr.it; luca.consolini@unipr.it; mattia.laurini@unipr.it; marco.locatelli@unipr.it).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3172169>.

Digital Object Identifier 10.1109/TAC.2022.3172169

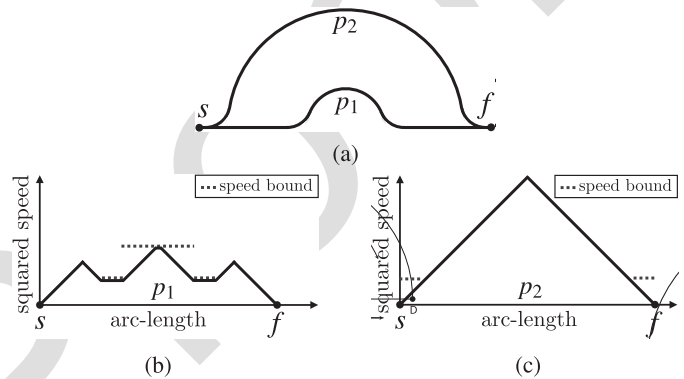


Fig. 1. Comparison of BASP and SP solutions. (a) Paths p_1 and p_2 connecting node s and f . (b) Optimal speed profile on p_1 . (c) Optimal speed profile on p_2 .

to illustrate the BASP and its difference with SP; it shows two fixed paths p_1 and p_2 connecting positions s and f . The vehicle starts from s with null speed and must reach f with null speed. The solution of SP corresponds to the path p_1 , which is the one of the shortest length. The BASP consists in finding the shortest-time path under acceleration and speed constraints. In this case, we assume that the vehicle acceleration and deceleration are bounded by a common constant and that its speed is bounded only on the central, high-curvature section of p_1 , in order to avoid excessive lateral acceleration, which may cause sideslip. If the bound on acceleration and deceleration is sufficiently high, the solution of the BASP corresponds to the path p_2 . Indeed, even if the latter path is longer, it can be traveled with a greater mean speed. Fig. 1(b) represents the fastest speed profile on p_1 . The x -axis corresponds to the arc-length position on the path p_1 and the y -axis represents the squared speed. In this representation, arc-length intervals of constant acceleration or deceleration correspond to straight lines. Fig. 1(c) represents the fastest speed profile on p_2 . Even if path p_2 is longer than p_1 , it can be traveled in less time. In fact, the vehicle is able to accelerate till the midpoint, and then, to decelerate to the end position f .

The interest for the BASP comes from a specific industrial application, namely the optimization of automated guided vehicles (AGVs) motion in automated warehouses. The AGVs may be either free to move within a facility or be only allowed to move along predetermined paths. In the first case, one needs to employ environmental representations such as cell decomposition methods [1] or trajectory maps [2]. In particular, the authors in [3] present an algorithm based on a modification of Dijkstra's algorithm in which edge weights are history dependent. Our work is related to the second approach. Namely, we assume that AGVs

cannot move freely within their environment and are instead required to move along predetermined paths that connect fixed operating points. These may be associated to shelves locations, where packages are stored or retrieved, to the end of production lines, where AGVs pick up final products, and to additional intermediate locations, used for routing. All these points are formally represented as nodes of a graph, whose arcs represent connecting paths. If AGVs are not subject to acceleration and speed constraints, the minimum-time planning problem is equivalent to SP and can be solved by the Dijkstra algorithm or its variants: see, for instance, [4]–[6], or other algorithms such as A* [7], Lifelong planning A* [8], D* [9], and D* Lite [10]. However, since the motion of AGVs must satisfy constraints on maximum speed and tangential and transversal accelerations that depend on the vehicle position on the path, these approaches cannot be applied to solve the BASP.

Instead, various works consider the minimum-time speed planning problem with acceleration and speed constraint on an *assigned* path. For instance, one can use the methods presented in [11] and [12], or path-following techniques such as [13] and [14].

As said, despite the fact that a large literature exists on SP and on the minimum-time speed planning on an assigned path, to the authors' knowledge, the BASP has never been specifically addressed in the literature. Formally, the BASP can be framed as an optimal control problem for a switching system, in which switchings are associated to passages from arc to arc and each discrete state is associated to a specific set of constraints. The results presented in this article exploit the very specific structure of the BASP and cannot be applied to generic switching systems. Anyway, the Algorithm V.5 could still apply to other switching systems satisfying an analogous of Proposition IV.3 and identifying a class of such systems could be the topic of future research.

This article is structured as follows. After introducing the notation employed throughout this article in Section II, in Section III, we first briefly discuss the solution of the speed planning problem along a *fixed* path, and then, we provide a formal statement of the BASP, also mentioning an NP-hardness result. In Section IV, we consider a subclass of the BASP, called k -BASP, which can be solved with polynomial time complexity for fixed values of k . Since constant k is problem dependent and is not known in advance, in Section V, we present an adaptive A* algorithm to find k . Finally, Section VI presents different computational experiments.

II. NOTATION

A directed graph is a pair $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where \mathbb{V} is a set of nodes and $\mathbb{E} \subset \{(x, y) \in \mathbb{V}^2 \mid x \neq y\}$ is a set of directed arcs. A path p on \mathbb{G} is a sequence of adjacent nodes of \mathbb{V} (i.e., $p = \sigma_1 \cdots \sigma_m$, with $(\forall i \in \{1, \dots, m\}) (\sigma_i, \sigma_{i+1}) \in \mathbb{E}$). An alphabet $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ is a set of symbols. A word is any finite sequence of symbols. The set of all words over Σ is Σ^* , which also contains the empty word ε , while Σ_i represents the set of all words of length up to $i \in \mathbb{N}$, (i.e., words composed of up to i symbols, including ε). Given a word $w \in \Sigma^*$, $|w|$ denote its length. Given a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, we can think of \mathbb{V} as an alphabet so that any path p of \mathbb{G} is a word in \mathbb{V}^* . Given $s, t \in \Sigma^*$, the word obtained by writing t after s is the concatenation of s and t , denoted by $st \in \Sigma^*$; we call t a suffix of st and s a prefix of st . For $r \in \Sigma^*$, \bar{r} is the rightmost symbol of r . In the following, we represent paths of \mathbb{G} as strings of symbols in \mathbb{V} . This allows to use the concatenation operation on paths and to use prefixes and suffixes to represent portions of paths. For $x \in \mathbb{R}$, $\lceil x \rceil = \min\{i \in \mathbb{Z} \mid i \geq x\}$ is the ceiling of x . For $a, b \in \mathbb{R}$, we set $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$, as the minimum and maximum operations, respectively.

Finally, given an interval $I \subseteq \mathbb{R}$, we recall that $W^{1,\infty}(I)$ is the Sobolev space of functions in $L^\infty(I)$ with weak derivative of order 1 with finite L^∞ -norm. For $f, g \in W^{1,\infty}(I)$, we denote with $f \wedge g$ and $f \vee g$ the point-wise minimum and maximum of f and g , respectively.

III. PROBLEM FORMULATION

Before giving the formal description of the BASP, in Section III-A, we briefly discuss the solution of the speed planning problem along a fixed path. Although such problem has been already widely discussed in the literature, here, we briefly describe a way to tackle it in order to better understand the following formulation of the BASP.

A. Speed Planning Along an Assigned Path

Let $\gamma : [0, \lambda_f] \rightarrow \mathbb{R}^2$ be a C^2 function such that $(\forall \lambda \in [0, \lambda_f]) \|\gamma'(\lambda)\| = 1$. The image set $\gamma([0, \lambda_f])$ represents the path followed by a vehicle, $\gamma(0)$ the initial configuration, and $\gamma(\lambda_f)$ the final one. The function γ is an arc-length parameterization of a path. We want to compute the speed law that minimizes the overall travel time while satisfying some kinematic and dynamic constraints. To this end, let $\xi : [0, t_f] \rightarrow [0, \lambda_f]$ be a differentiable monotonically increasing function representing the vehicle arc-length coordinate along the path as a function of time and let $v : [0, \lambda_f] \rightarrow [0, +\infty)$ be such that $(\forall t \in [0, t_f]) \dot{\xi}(t) = v(\xi(t))$. In this way, $v(\lambda)$ is the vehicle speed at position λ . The vehicle position as a function of time is given by $x : [0, t_f] \rightarrow \mathbb{R}^2$, $x(t) = \gamma(\xi(t))$, speed and acceleration are given by $\dot{x}(t) = \gamma'(\xi(t))v(\xi(t))$, and $\ddot{x}(t) = a_L(t)\gamma'(\xi(t)) + a_N(t)\gamma'^{\perp}(\xi(t))$, where $a_L(t) = v'(\xi(t))v(\xi(t))$ and $a_N(t) = \kappa(\xi(t))v(\xi(t))^2$ are the longitudinal and normal components of acceleration, respectively. Here, $\kappa : [0, \lambda_f] \rightarrow \mathbb{R}$ is the scalar curvature, defined as $\kappa(\lambda) = \langle \gamma''(\lambda), \gamma'(\lambda)^\perp \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the scalar product.

We require to travel distance λ_f in a minimum time while satisfying, for every $t \in [0, \xi^{-1}(\lambda_f)]$, $0 \leq v^-(\xi(t)) \leq v(\xi(t)) \leq v^+(\xi(t))$, $|a_N(\xi(t))| \leq \beta(\xi(t))$, $\alpha^-(\xi(t)) \leq a_L(\xi(t)) \leq \alpha^+(\xi(t))$. Here, functions v^- , v^+ , α^- , α^+ , and β are arc-length-dependent bounds on the vehicle speed and on its longitudinal and normal acceleration. It is convenient to make the change of variables $w = v^2$ (see [15]) so that by setting $\Psi(w) = \int_0^{\lambda_f} w(\lambda)^{-\frac{1}{2}} d\lambda$, $\mu^+(\lambda) = v^+(\lambda)^2 \wedge \frac{\beta(\lambda)}{\kappa(\lambda)}$, and $\mu^-(\lambda) = v^-(\lambda)^2$, our problem takes on the following form.

$$\min_{w \in W^{1,\infty}([0, \lambda_f])} \Psi(w) \quad (1a)$$

$$\mu^-(\lambda) \leq w(\lambda) \leq \mu^+(\lambda), \quad \lambda \in [0, \lambda_f] \quad (1b)$$

$$\alpha^-(\lambda) \leq w'(\lambda) \leq \alpha^+(\lambda), \quad \lambda \in [0, \lambda_f] \quad (1c)$$

where $\Psi : W^{1,\infty}([0, \lambda_f]) \rightarrow \mathbb{R}$ is order reversing (i.e., $(\forall x, y \in [0, \lambda_f]) x \geq y \Rightarrow \Psi(x) \leq \Psi(y)$) and μ^- , μ^+ , α^- , $\alpha^+ \in L^\infty([0, \lambda_f])$ are assigned functions with $\mu^-, \alpha^+ \geq 0$, and $\alpha^- \leq 0$. Initial and final conditions on speed can be included in the definition of functions μ^- and μ^+ . For instance, to set initial condition $w(0) = w_0$, it is sufficient to define $\mu^+(0) = \mu^-(0) = w_0$. In [16], we introduced a subset of $W^{1,\infty}([0, \lambda_f])$, called Q , as a technical requirement and an operator based on the solution of the following differential equations:

$$\begin{cases} F'(\lambda) = \begin{cases} \alpha^+(\lambda) \wedge \mu'(\lambda), & \text{if } F(\lambda) \geq \mu(\lambda) \\ \alpha^+(\lambda), & \text{if } F(\lambda) < \mu(\lambda) \end{cases} \\ F(0) = \mu(0) \end{cases} \quad (2)$$

$$\begin{cases} B'(\lambda) = \begin{cases} \alpha^-(\lambda) \wedge \mu'(\lambda), & \text{if } B(\lambda) \geq \mu(\lambda) \\ \alpha^-(\lambda), & \text{if } B(\lambda) < \mu(\lambda) \end{cases} \\ B(\lambda_f) = \mu(\lambda_f) \end{cases} \quad (3)$$

with $F, B \in Q$, that allows to compute the optimal solution of the Problem (1). In particular, in [16], it is shown that the optimal solution is $F(\mu^+) \wedge B(\mu^+)$. We refer the reader to [16] for a detailed discussion.

B. BASP Problem

In this section, we provide a formal description of the BASP. Let us consider a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, with $\mathbb{V} = \{\sigma_1, \dots, \sigma_N\}$. For each $i \in \{1, \dots, N\}$, the node σ_i represents an operating point $R_i \in \mathbb{R}^2$. In fact, the restriction $R_i \in \mathbb{R}^2$ is not strictly necessary but we imposed it since it holds in the AGV application, which is the main motivation of this work. Each arc $\theta = (\sigma_i, \sigma_j) \in \mathbb{E}$ represents a fixed directed path between two operating points and is associated to an arc-length parameterized path γ_θ of length $\ell(\theta)$, such that $\gamma_\theta(0) = R_i$ and $\gamma_\theta(\ell(\theta)) = R_j$. In the following, we denote the set of all possible paths on \mathbb{G} by P . Similarly, for $s, f \in \mathbb{V}$, we denote by P_s the subset of P consisting in all paths starting from s and by $P_{s,f}$ the subset of P consisting in all paths starting from s and ending in f . We extend this definition to subsets of \mathbb{V} , that is, if $S, F \subset \mathbb{V}$, we denote by $P_{S,F}$ the set of all paths starting from nodes in S and ending in nodes in F . Given a path $p = \sigma_1 \cdots \sigma_m$, its length $\ell(p)$ is defined as the sum of the lengths of its individual arcs, that is, $\ell(p) = \sum_{i=1}^{m-1} \ell(\sigma_i, \sigma_{i+1})$.

To setup our problem, we need to associate four real-valued functions to each edge $\theta \in \mathbb{E}$: the maximum and minimum allowed acceleration and squared speed. The domain of each function is the arc-length coordinate on the path γ_θ . Then, given a specific path p , we are able to define a speed optimization problem of the form (1) by considering the constraints associated to the edges that compose p . We define the set of edge functions as $\mathcal{E} = \{\varphi : \mathbb{E} \times \mathbb{R}^+ \rightarrow \mathbb{R}\}$. If $\varphi \in \mathcal{E}$, $\theta \in \mathbb{E}$, $\lambda \in \mathbb{R}^+$, $\varphi(\theta, \lambda)$ denotes the value of φ on edge θ at position λ . Note that $\varphi(\theta, \lambda)$ will be relevant only for $\lambda \in [0, \ell(\theta)]$. Given a path $p = \sigma_1 \cdots \sigma_m$, we associate to $\varphi \in \mathcal{E}$ a function $\varphi_p : [0, \ell(p)] \rightarrow \mathbb{R}$ in the following way. Define functions $\Theta : [0, \ell(p)] \rightarrow \mathbb{N}$, $\Lambda : [0, \ell(p)] \rightarrow \mathbb{R}$ such that $\Theta(\lambda) = \max\{i \in \mathbb{N} \mid \ell(\sigma_1 \cdots \sigma_i) \leq \lambda\}$ and $\Lambda(\lambda) = \ell(\sigma_1 \cdots \sigma_{\Theta(\lambda)})$. In this way, $\Theta(\lambda)$ is such that $\theta(\lambda) = (\sigma_{\Theta(\lambda)}, \sigma_{\Theta(\lambda)+1})$ is the edge that contains the position at arc length λ along p , and $\Lambda(\lambda)$ is the sum of the lengths of all arcs up to node $\sigma_{\Theta(\lambda)}$ in p . Then, we define $\varphi_p(\lambda) = \varphi(\theta(\lambda), \lambda - \Lambda(\lambda))$.

Given $\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^+, \hat{\alpha}^- \in \mathcal{E}$ and path $p \in P$, let $\mathbb{B} = (\hat{\mu}^-, \hat{\mu}^+, \hat{\alpha}^-, \hat{\alpha}^+)$. Assume $(\forall \theta \in \mathbb{E}) \hat{\mu}^+(\theta, \cdot) \in Q$ and define $T_{\mathbb{B}}(p) = \min_{w \in W^{1,\infty}([0, \ell(p)])} \Psi(w)$, as the solution of the Problem (1) along path p with $\mu^- = \hat{\mu}^-$, $\mu^+ = \hat{\mu}^+$, $\alpha^- = \hat{\alpha}^-$, and $\alpha^+ = \hat{\alpha}^+$. In this way, $T_{\mathbb{B}}(p)$ is the minimum time required to traverse the path p , respecting the speed and acceleration constraints defined in \mathbb{B} . We set $T_{\mathbb{B}}(p) = +\infty$ if the Problem (1) is not feasible.

The following is the main problem discussed in this article.

Problem III.1 (BASP): Given a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, $\mu^+, \mu^-, \alpha^-, \alpha^+ \in \mathcal{E}$, $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+)$, $s \in \mathbb{V}$, and $F \subset \mathbb{V}$, find $\min_{p \in P_{s,F}} T_{\mathbb{B}}(p)$.

In other words, we want to find the path p that minimizes the transfer time between source node s and a destination node in F , taking into account bounds on speed and accelerations on each traversed arc (represented by arc functions $\mu^+, \mu^-, \alpha^-, \alpha^+$). The following properties are a direct consequence of the definition of $T_{\mathbb{B}}(p)$.

Proposition III.2: The following properties hold:

- 1) let $p_1, p_2 \in P$, $p_1 p_2 \in P \Rightarrow T_{\mathbb{B}}(p_1 p_2) \geq T_{\mathbb{B}}(p_1) + T_{\mathbb{B}}(p_2)$;
- 2) if $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+)$, $\hat{\mathbb{B}} = (\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^-, \hat{\alpha}^+)$ are such that $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) [\mu^-(\theta, \lambda), \mu^+(\theta, \lambda)] \subset [\hat{\mu}^-(\theta, \lambda), \hat{\mu}^+(\theta, \lambda)]$

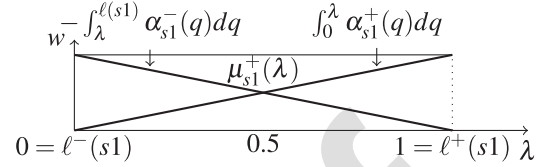


Fig. 2. Computation of $\ell^+(s1) = 1$ and $\ell^-(s1) = 0$.

and $[\alpha^-(\theta, \lambda), \alpha^+(\theta, \lambda)] \subset [\hat{\alpha}^-(\theta, \lambda), \hat{\alpha}^+(\theta, \lambda)]$, then $(\forall p \in P) T_{\mathbb{B}}(p) \geq T_{\hat{\mathbb{B}}}(p)$.

In particular, the first property states that the minimum time for traveling the composite path $p_1 p_2$ is greater or equal to the sum of the times needed for traveling p_1 and p_2 separately. In fact, in the first case, the speed must be continuous when passing from p_1 to p_2 (due to the acceleration bounds), but this constraint does not need to be satisfied when the speed profiles for p_1 and p_2 are computed separately.

The following proposition (whose proof can be found in [17]) states the theoretical complexity of a simplified version of Problem III.1, called BASP-C, in which maximum and minimum acceleration and speed are constant on each arc.

Proposition III.3: Problem BASP-C is NP-hard.

IV. k-BASP

As we will see in Remark IV.6, SP can be viewed as a special case of the BASP, namely a BASP with unbounded acceleration limits. In fact, also BASP can be viewed as an SP but defined on a different graph with respect to the original one. More precisely, here, we introduce some restrictions on parameters \mathbb{B} that allow reducing the BASP to a standard SP that can be solved by Dijkstra's algorithm on an extended graph. Let $p \in P$, define

$$\ell^+(p) = \min\{\lambda \in [0, \ell(p)] \mid \int_0^\lambda \alpha_p^+(q) dq = \mu_p^+(\lambda), +\infty\};$$

$$\ell^-(p) = \max\{\lambda \in [0, \ell(p)] \mid -\int_\lambda^{\ell(p)} \alpha_p^-(q) dq = \mu_p^-(\lambda), -\infty\}.$$

In this way, $\ell^+(p)$ is the smallest value of $\lambda \in [0, \ell(p)]$ for which the solution of F in (2), with $\alpha^+ = \alpha_p^+$, starting from initial condition $F(0) = 0$, reaches the squared speed upper bound $\mu^+(\lambda)$. Note that $\ell^+(p) = \infty$ if no such value of λ exists. Similarly, $\ell^-(p)$ is the largest value of $\lambda \in [0, \ell(p)]$ for which the solution of B in (3), with $\alpha^- = \alpha_p^-$, starting from initial condition $B(\ell(p)) = 0$, reaches $\mu^+(\lambda)$. Again, $\ell^-(p) = -\infty$ if no such value of λ exists. Note that if $p, t, pt \in P$, $\ell^+(pt) \leq \ell^+(p)$ and $\ell^-(pt) \geq \ell^-(p)$ (actually, equalities hold if the values are all finite). Finally, we define

$$K(\mathbb{B}) = \min\{k \in \mathbb{N} \mid (\forall p \in P_s) |p| \geq k \Rightarrow \ell^+(p) \leq \ell^-(p)\}. \quad (4)$$

We call k -BASP any instance of Problem III.1 that satisfies $K(\mathbb{B}) \leq k$. For instance, consider the following chain graph $\mathbb{G} = (\mathbb{V} = \{s, 1, 2, f\}, \mathbb{E} = \{(s, 1), (1, 2), (2, f)\})$. Here, $(\forall \theta \in \mathbb{E}) \alpha^-(\theta) = -1$, $\alpha^+(\theta) = 1$, $\mu^-(\theta) = 0$, $\ell(\theta) = 1$, and $\mu^+((s, 1)) = 1$, $\mu^+((1, 2)) = \frac{2}{3}$, $\mu^+((2, f)) = 1$. In this case, $P_s = \{s, s1, s12, s12f\}$. Moreover, $K(\mathbb{B}) > 2$, since $\ell^+(s1) = 1 > 0 = \ell^-(s1)$, as reported in Fig. 2. Furthermore, $\ell^+(s12) < \ell^-(s12)$ and $\ell^+(12f) < \ell^-(12f)$ and $s12, 12f$ are the only paths of length 3. Fig. 3 shows the computation of $\ell^+(s12)$ and $\ell^-(s12)$; the computation of $\ell^+(12f)$ and $\ell^-(12f)$ is analogous. Hence, in this example, $K(\mathbb{B}) = 3$.

Note that $K(\mathbb{B}) - 1$ represents the maximum number of nodes of a path that can be traveled with a speed profile of maximum acceleration, followed by one of maximum deceleration, starting and ending with null speed, without violating the maximum speed constraint. The following

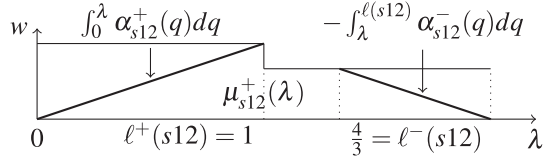


Fig. 3. Computation of $\ell^+(s12) = 1$ and $\ell^-(s12) = \frac{4}{3}$.

273 expression provides a simple upper bound on $K(\mathbb{B})$:

$$K(\mathbb{B}) \leq 1 + \left[2 \max_{\theta \in \mathbb{E}} \frac{\max_{\lambda \in [0, \ell(\theta)]} \mu^+(\theta, \lambda)}{\min_{\lambda \in [0, \ell(\theta)]} (\alpha^+(\theta, \lambda) \wedge |\alpha^-(\theta, \lambda)|) \ell(\theta)} \right]. \quad (5)$$

274 Note that $K(\mathbb{B}) = 1$ only if $\alpha_- = -\infty$ and $\alpha^+ = +\infty$, that is, if we
 275 do not consider acceleration bounds. We will present an algorithm that
 276 solves the k -BASP in polynomial time complexity with respect to $|\mathbb{V}|$
 277 and $|\mathbb{E}|$. However, note that the complexity is exponential with respect
 278 to k so that a correct estimation of $K(\mathbb{B})$ is critical. In general, the
 279 bound (5) overestimates $K(\mathbb{B})$. In Section V, we will present a simple
 280 method for correctly estimating $K(\mathbb{B})$.

281 We recall that \mathbb{V}_k represents the subset of language \mathbb{V}^* composed of
 282 strings with maximum length k , including the empty string ε . Define
 283 $\text{Suff}_k : P \rightarrow \mathbb{V}_k$ such that, if $|p| \leq k$, $\text{Suff}_k(p) = p$ and if $|p| > k$,
 284 $\text{Suff}_k(p)$ is the suffix of p of length k . The function Suff_k allows to
 285 introduce a partially defined transition function $\Gamma : \mathbb{V}_k \times \mathbb{V} \rightarrow \mathbb{V}_k$ by
 286 setting $\Gamma(r, \sigma) = \text{Suff}_k(r\sigma)$ if $r\sigma \in P$, otherwise, if $r\sigma \notin P$, $\Gamma(r, \sigma)$
 287 is not defined. Define the incremental cost function $\eta : P_s \times \mathbb{V} \rightarrow \mathbb{R}^+$
 288 such that, for $p \in P_s$ and $\sigma \in \mathbb{V}$, if $p\sigma \in P_s$, $\eta(p, \sigma) = T_{\mathbb{B}}(p\sigma) -$
 289 $T_{\mathbb{B}}(p)$, otherwise $\eta(p, \sigma) = +\infty$. In other words, $\eta(p, \sigma)$ is the dif-
 290 ference between the minimum time required for traversing $p\sigma$ and the
 291 minimum time required for traversing p . For simplicity of notation,
 292 from now on, we will denote $T_{\mathbb{B}}$ simply as T . The following proposition
 293 shows that the incremental cost is always strictly positive.

294 *Proposition IV.1:* $\eta(p, \sigma) \geq T(\sigma)$.

295 *Proof:* By 1) of Proposition III.2, $T(p\sigma) \geq T(p) + T(\sigma)$. \square

296 The following property, whose proof is presented in the Appendix,
 297 plays a key role in the solution algorithm.

298 *Proposition IV.2:* Let $p_1, p_2, t \in P$, if $p_1t, p_2t \in P$ and $\ell^+(t) \leq$
 299 $\ell^-(t)$, then $(\forall \sigma \in \mathbb{V}) T(p_1t\sigma) - T(p_1t) = T(p_2t\sigma) - T(p_2t)$.

300 The following is a direct consequence of Proposition IV.2. It states
 301 that, given $p \in P$ and $\sigma \in \mathbb{V}$, the incremental cost $\eta(p, \sigma)$ does not
 302 depend on the complete path p , but only on $\text{Suff}_k(p)$ (its last k symbols).

303 *Proposition IV.3:* If $K(\mathbb{B}) \leq k$ and $p, p' \in P$ are such that
 304 $\text{Suff}_k(p) = \text{Suff}_k(p')$, then $(\forall \sigma \in \mathbb{V}) \eta(p, \sigma) = \eta(p', \sigma)$.

305 Define function $\hat{\eta} : \mathbb{V}_k \times \mathbb{V} \rightarrow \mathbb{R}^+$, such that $\hat{\eta}(r, \sigma) = \eta(p, \sigma)$
 306 where $p \in P$ is any path such that $r = \text{Suff}_k(p)$. We set $\hat{\eta}(r, \sigma) = +\infty$
 307 if such path does not exist. Note that the function $\hat{\eta}$ is well-defined by
 308 Proposition IV.3, being $\eta(p, \sigma)$ identical among all paths p such that $r =$
 309 $\text{Suff}_k(p)$. In particular, Proposition IV.3 holds for $p' = \text{Suff}_k(p) = r$
 310 so that we can compute $\hat{\eta}$ as $\hat{\eta}(r, \sigma) = \eta(r, \sigma)$. In the following, since
 311 $\hat{\eta}$ is the restriction of η on $\mathbb{V}_k \times \mathbb{V}$, we denote $\hat{\eta}$ simply by η .

312 The value k can be viewed as the amount of memory required to
 313 solve the problem: once a node is reached, the optimal path from such
 314 node to the target one depends on the last k visited nodes. If $k = 1$, it
 315 only depends on the current node (i.e., no memory is required). This
 316 is the situation with the classical SP. More generally, $k > 1$ so that the
 317 optimal way to complete the path does not only depend on the current
 318 node, but also on the sequence of $k - 1$ nodes visited before reaching
 319 it. Define function $V : \mathbb{V}_k \rightarrow \mathbb{R}$ as

$$V(r) = \min_{p \in P_s | \text{Suff}_k p = r} T_{\mathbb{B}}(p). \quad (6)$$

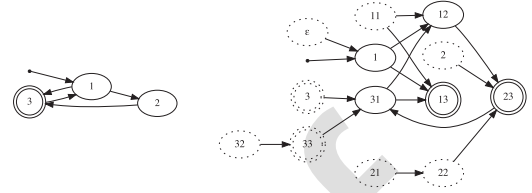


Fig. 4. Graph and its corresponding extension for $k = 2$.

Note that the solution of the BASP corresponds to $\min_{r \in \mathbb{V}_k | \tilde{r} \in F} V(r)$ 320
 (we recall that \tilde{r} is the last node of r). For $r \in \mathbb{V}_k$, define the set of 321
 predecessors of r as $\text{Prec}(r) = \{\tilde{r} \in \mathbb{V}_k \mid r = \Gamma(\tilde{r}, \tilde{r})\}$. The following 322
 proposition presents an expression for $V(r)$ that holds if $\ell^+(r') \leq$ 323
 $\ell^-(r')$ for all predecessors r' of r . 324

325 *Proposition IV.4:* Let $r \in \mathbb{V}_k$, if $(\forall r' \in \text{Prec}(r)) \ell^+(r') \leq \ell^-(r')$,
 326 then

$$V(r) = \min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r', \tilde{r})\}. \quad (7)$$

327 *Proof:* Let $S_r = \{q \in P_s \mid \text{Suff}_k q\tilde{r} = r\}$. $V(r) = \min_{p \in$ 328
 $P_s \mid \text{Suff}_k p = rT(p) = \min_{q \in S_r} \{T(q\tilde{r}) - T(q) + T(q)\} = \min_{q \in$ 329
 $S_r} \{T(q) + T((\text{Suff}_k q)\tilde{r}) - T(\text{Suff}_k q)\} = \min_{q \in S_r} \{T(q) +$ 330
 $\eta(\text{Suff}_k q, \tilde{r})\} = \min_{r' \in \text{Prec}(r), q \in S_{r'}} \{T(q) + \eta(r', \tilde{r})\} =$ 331
 $\min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r', \tilde{r})\}$, where we used the facts that 332
 $T(q\sigma) - T(q) = T(\text{Suff}_k q\sigma) - T(\text{Suff}_k q)$, by Proposition IV.2, 333
 and that $q \in P_s$ is such that $\text{Suff}_k q\tilde{r} = r \Leftrightarrow \text{Suff}_k q \in \text{Prec}(r)$. \blacksquare 334

335 As a consequence of Proposition IV.4, if $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$,
 336 $V(r)$ corresponds to the length of the shortest path from s to r on the
 337 extended directed graph $\tilde{\mathbb{G}} = (\tilde{\mathbb{V}}, \tilde{\mathbb{E}})$, where $\tilde{\mathbb{V}} = \mathbb{V}_k$ and $(r_1, r_2) \in \tilde{\mathbb{E}}$ 338
 if $r_2 = \Gamma(r_1, \tilde{r}_2)$ is defined, in this case its length is $\eta(r_1, \tilde{r}_2)$. The left 339
 part of Fig. 4 shows a graph consisting of three nodes. Node $s = 1$ is 340
 the source (indicated by the entering arrow) and the double border 341
 shows the final node $F = \{3\}$. The right part of Fig. 4 represents 342
 the corresponding extended graph, obtained for $k = 2$, consisting of 343
 13 nodes (the cardinality of \mathbb{V}_2). Note that some of the nodes are 344
 unreachable from the initial state, these are represented with dotted 345

346 borders. Solving k -BASP corresponds to finding a minimum-length path on 347
 $\tilde{\mathbb{G}}$ that connects node $s \in \mathbb{V}_k$ to $\tilde{F} = \{r \in \mathbb{V}_k \mid \tilde{r} \in F\}$. Note that the 348
 set of final states \tilde{F} for the extended graph $\tilde{\mathbb{G}}$ contains all paths $p \in \mathbb{V}_k$ 349
 that end in an element of F . In the extended graph reported in Fig. 4, this 350
 corresponds to finding a minimum-length path from the starting node 351
 1 to one of the final nodes 3, 13, 23, and 33. Note that the unreachable 352
 nodes play no role in this procedure. We can find a minimum-length 353
 path by Dijkstra's algorithm applied to $\tilde{\mathbb{G}}$, leading to the following 354
 complexity result. 355

356 *Proposition IV.5:* k -BASP can be solved with complexity
 $O(|\mathbb{V}|^{k-1}|\mathbb{E}| + (|\mathbb{V}|^k \log |\mathbb{V}|^k))$. 357

358 *Proof:* Dijkstra's algorithm has time complexity $O(|E| +$
 $|\mathbb{V}| \log |\mathbb{V}|)$, where $|E|$ and $|\mathbb{V}|$ are the cardinalities of the edge 359
 and vertex sets, respectively. In our case, $|\mathbb{V}| = |\tilde{\mathbb{V}}| = |\mathbb{V}_k| =$
 $\sum_{i=0}^k |\mathbb{V}|^i = O(|\mathbb{V}|^k)$, $|E| = |\tilde{\mathbb{E}}| \leq |\mathbb{V}_{k-1}|\mathbb{E}| = O(|\mathbb{V}|^{k-1}|\mathbb{E}|)$. \square 360

361 The following remark establishes that SP can be viewed as a special
 case of the BASP without acceleration bounds.

362 *Remark IV.6:* If $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) \alpha^-(\theta, \lambda) = -\infty,$
 363 $\alpha^+(\theta, \lambda) = +\infty$, then $K(\mathbb{B}) = 1$. The resulting 1-BASP reduces to
 a standard SP on the graph \mathbb{G} and can be solved with time complexity 364
 $O(|\mathbb{E}| + |\mathbb{V}| \log |\mathbb{V}|)$. 365

V. ADAPTIVE A* ALGORITHM FOR k -BASP 366

367 The computation method based on Dijkstra's algorithm on the
 extended graph $\tilde{\mathbb{G}}$, presented in the previous section, has two main 368

disadvantages. First, $\tilde{\mathbb{G}}$ has $\sum_{j=0}^k |\mathbb{V}|^j$ nodes so that the time required by Dijkstra's algorithm grows exponentially with k . We will show that it is possible to mitigate this problem and reduce the number of visited nodes by using the A* algorithm with a suitable heuristic. Second, the estimation of $k = K(\mathbb{B})$ from its definition is not an easy task. We will show that it is quite easy to adaptively find the correct value of k by starting from $k = 2$ and increasing k if needed.

The A* algorithm is a heuristic method that allows to compute the optimal path, if it exists (see [18]), by exploring the graph beginning from the starting node along the most promising directions according to a heuristic function that estimates the cost from the current position to the target node. Hence, to implement the A* algorithm, we need to define a heuristic function $h : \mathbb{V}_k \rightarrow \mathbb{R}$, such that, for $r \in \mathbb{V}_k$, $h(r)$ is a lower bound on $\min_{p \in P_{\tilde{r}, \tilde{F}}} T(p)$, that is, the minimum time needed for traveling from \tilde{r} to a final state in \tilde{F} . In general, we can compute lower bounds for the BASP by relaxing the acceleration constraints α^- and α^+ . Namely, let \mathbb{B} be a parameter set obtained by relaxing acceleration constraints in \mathbb{B} . Then, if $K(\mathbb{B}) < K(\mathbb{B})$, by Proposition IV.5, the solution of the BASP for parameter \mathbb{B} can be computed with a lower computational time than the solution with parameter \mathbb{B} . In particular, we obtain a very simple lower bound by removing acceleration bounds altogether, that is, by setting $\alpha^- = -\infty$ and $\alpha^+ = +\infty$. In this way, the vehicle is allowed to travel at maximum speed everywhere along the path and the incremental cost function $\eta(p, \sigma)$ is given by the time needed to travel γ_σ at maximum speed, that is, $\eta(p, \sigma) = \int_0^{\ell(\tilde{p}\sigma)} \frac{1}{\sqrt{\mu^+((\tilde{p}, \sigma), \lambda)}} d\lambda$.

Define the heuristic $h : \mathbb{V}_k \rightarrow \mathbb{R}^+$ as

$$h(r) = \min_{p \in P_{\tilde{r}, \tilde{F}}} T_{\mathbb{B}}(p). \quad (8)$$

Note that, if $\alpha^- = -\infty$ and $\alpha^+ = +\infty$, h corresponds to the solution of 1-BASP and all values of h can be efficiently precomputed by Dijkstra's algorithm (see Remark IV.6). The following proposition shows that h is admissible and consistent so that the A* algorithm, with heuristic h , provides the optimal solution of the k -BASP and its time complexity is no worse than Dijkstra's algorithm (see [19, Th. 2.9 and 2.10]).

Proposition V.1: Heuristic h satisfies the following two properties.

- 1) Admissibility: $(\forall r \in \mathbb{V}_k) h(r) \leq \min_{q \in P_{\tilde{r}, \tilde{F}}} T_{\mathbb{B}}(q)$.
- 2) Consistency: $(\forall r \in \mathbb{V}_k) (\forall \sigma \in \mathbb{V}) h(r) \leq \eta(r, \sigma) + h(\Gamma(r, \sigma))$.

Proof: 1) $h(r) = \min_{p \in P_{\tilde{r}, \tilde{F}}} T_{\mathbb{B}}(p) \leq \min_{q \in P_{\tilde{r}, \tilde{F}}} T_{\mathbb{B}}(q)$, since \mathbb{B} is a relaxation of \mathbb{B} .

2) $h(r) = \min_{p \in P_{\tilde{r}, \tilde{F}}} T_{\mathbb{B}}(p) \leq T_{\mathbb{B}}(\sigma) + \min_{p \in P_{\sigma, \tilde{F}}} T_{\mathbb{B}}(p) \leq T_{\mathbb{B}}(\sigma) + \min_{p \in P_{\sigma, \tilde{F}}} T_{\mathbb{B}}(p) \leq \eta(r, \sigma) + h(\Gamma(r, \sigma))$, where $T_{\mathbb{B}}(\sigma) \leq T_{\mathbb{B}}(\sigma)$ by 2) of Proposition III.2 and $T_{\mathbb{B}}(\sigma) \leq \eta(r, \sigma)$ by Proposition IV.1. \square

Since heuristic h is admissible and consistent, A* is equivalent to Dijkstra's algorithm, with the only difference that the incremental cost function $\eta(r, \sigma)$ is replaced by the modified cost

$$\tilde{\eta}(r, \sigma) = \eta(r, \sigma) + h(\Gamma(r, \sigma)) - h(r) \quad (9)$$

(see [19, Lemma 2.3] for a complete discussion). A description of the A* algorithm can be found in literature (for instance, see [19, Algorithm 2.13]). We define a priority queue \mathcal{Q} that contains open nodes, that is, nodes that have already been generated but have not yet been visited. Namely, \mathcal{Q} is an ordered set of pairs $(r, t) \in \mathbb{V}_k \times \mathbb{R}^+$, in which $r \in \mathbb{V}_k$ and t is a lower bound for the time associated to the best completion of r to a path arriving at a final state. We need to perform the following operations on \mathcal{Q} : operation $\text{Insert}(\mathcal{Q}, (r, t))$ inserts couple (r, t) into \mathcal{Q} ; operation $(r, t) = \text{DeleteMin}(\mathcal{Q})$ returns the first couple of \mathcal{Q} , that is, the couple (r, t) with the minimum time t , and removes this couple from \mathcal{Q} ; and, operation $\text{DecreaseKey}(\mathcal{Q}, (r, t))$ assumes that \mathcal{Q} already contains a couple (r, t') with $t' > t$ and substitutes this

couple with (r, t) . Furthermore, we consider three partially defined maps $\text{value} : \mathbb{V}_k \rightarrow \mathbb{R}$, $\text{parent} : \mathbb{V}_k \rightarrow \mathbb{V}_k$, $\text{closed} : \mathbb{V}_k \rightarrow \{0, 1\}$, such that, for $r \in \mathbb{V}_k$, $\text{value}(r)$ is the current best upper estimate of $V(r)$, $\text{parent}(r)$ is the parent node of r , and $\text{closed}(r) = 1$ if node r has already been visited. Maps value , parent , and closed can be implemented as hash tables.

Algorithm V.2 (A algorithm for k -BASP):*

1) [initialization] Set $\mathcal{Q} = \{(s, h(s))\}$, $\text{value}(s) = 0$.

2) [expansion] Set $(r, t) = \text{DeleteMin}(\mathcal{Q})$ and set $\text{closed}(r) = 1$. If $\tilde{r} \in \tilde{F}$, then t is the optimal solution and the algorithm terminates, returning maps value , parent . Otherwise, for each $\sigma \in \mathbb{V}$ for which $\Gamma(r, \sigma)$ is defined, set $r' = \Gamma(r, \sigma)$, $t' = t + \tilde{\eta}(r, \sigma)$. If $\text{closed}(r') = 1$, go to 3). Else, if $\text{value}(r')$ is undefined $\text{Insert}(\mathcal{Q}, (r', t'))$. Otherwise, if $t' < \text{value}(r')$, set $\text{value}(r') = t'$, $\text{parent}(r') = r$ and do $\text{DecreaseKey}(\mathcal{Q}, (r', t'))$.

3) [loop] If $\mathcal{Q} \neq \emptyset$ go back to 2), otherwise no solution exists.

Proposition V.3: Algorithm V.2 terminates and returns the optimal solution (if it exists), with a time-complexity not higher than Dijkstra's algorithm on the extended graph $\tilde{\mathbb{G}}$.

Proof: It is a consequence of the fact that heuristic h is admissible and consistent (see [19, Th. 2.9 and 2.10]). \square

Note that, at the end of Algorithm V.2, $\text{value}(f)$ is the optimal value of the k -BASP and the optimal path from s to set F can be reconstructed from map parent .

One possible limitation of Algorithm V.2 is that estimating $K(\mathbb{B})$ from its definition can be difficult. A correct estimation of $K(\mathbb{B})$ is critical for the efficiency of the algorithm. Indeed, if $K(\mathbb{B})$ is overestimated, the time complexity of the algorithm is higher than it would be with a correct estimate. On the other hand, if $K(\mathbb{B})$ is underestimated, Algorithm V.2 is not correct since Proposition IV.4 does not hold. Here, we propose an algorithm that adaptively finds a suitable value for k in Algorithm V.2, such that $k \leq K(\mathbb{B})$, but, in any case, allows to find the optimal solution of the BASP. First, we define the modified cost function $W : \mathbb{V}_k \rightarrow \mathbb{R}$ as $W(r) = V(r) + h(r)$, where V is given by (6) and h is the heuristic given by (8). If $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$, then W is the solution of

$$\begin{cases} W(r) = \min_{r' \in \text{Prec}(r)} \{W(r') + \tilde{\eta}(r, r')\} \\ W(s) = h(s). \end{cases} \quad (10)$$

Indeed, following the same steps of the proof of Proposition IV.4, $W(r) = V(r) + h(r) = \min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r, r') + h(r) + h(r') - h(r')\} = \min_{r' \in \text{Prec}(r)} \{W(r') + \tilde{\eta}(r, r')\}$. Hence, $W(r)$ corresponds to the length of the shortest path from s to r on $\tilde{\mathbb{G}}$, with arc length given according to $\tilde{\eta}$. If condition $\ell^+(r) \leq \ell^-(r)$ is not satisfied for all $r \in \mathbb{V}_k$, (10) does not hold for all $r \in \mathbb{V}_k$ and W does not represent the solution of an SP. However, the following proposition shows that we can still find a lower bound \hat{W} of W that does correspond to the solution of an SP.

Proposition V.4: Let $\hat{W} : \mathbb{V}_k \rightarrow \mathbb{R}$ be the solution of

$$\begin{cases} \hat{W}(r) = \min_{r' \in \text{Prec}(r)} \{\hat{W}(r') + \hat{\eta}(r, r')\} \\ \hat{W}(s) = 0, \end{cases} \quad (11)$$

where if $\ell^+(r') \leq \ell^-(r')$ or $|r'| < k$, $\hat{\eta}(r, r') = \tilde{\eta}(r, r')$, otherwise $\hat{\eta}(r, r') = h(r) - h(r')$. Then, $(\forall r \in \mathbb{V}_k)$

1) $\hat{W}(r) \leq W(r)$;

2) $(\forall \tilde{r} \in \mathbb{V}_k \mid \tilde{W}(\tilde{r}) \leq \hat{W}(\tilde{r})) \ell^+(\tilde{r}) \leq \ell^-(\tilde{r}) \Rightarrow \hat{W}(\tilde{r}) = W(\tilde{r})$.

Proof: 1) For $r \in \mathbb{V}_k$, let $p \in P_s$ be such that $\text{Suff}_k p \in \text{Prec}(r)$. If $\ell^+(\text{Suff}_k p) \leq \ell^-(\text{Suff}_k p)$, in view of Proposition IV.2, $T(p\tilde{r}) = T(p) + \eta(\text{Suff}_k p, \tilde{r})$, otherwise, obviously, $T(p\tilde{r}) \geq T(p)$. Hence, in both cases, by the definition of $\tilde{\eta}$ in (9), $T(p\tilde{r}) + h(r) \geq T(p) + h(\text{Suff}_k p) + \hat{\eta}(\text{Suff}_k p, \tilde{r})$. By contradiction, assume

480 $(\exists A \subset \mathbb{V}_k) A \neq \emptyset$ such that $(\forall r \in A) \hat{W}(r) > W(r)$. Let
 481 $\bar{r} = \operatorname{argmin}_{\bar{r} \in A} W(\bar{r})$ and $S_{\bar{r}} = \{q \in P_s \mid \operatorname{Suff}_k q \in \operatorname{Prec}(\bar{r})\}$,
 482 then $W(\bar{r}) = V(\bar{r}) + h(\bar{r}) = \min_{p \in P_s \mid \operatorname{Suff}_k p = \bar{r}} T(p) + h(\bar{r}) =$
 483 $\min_{q \in S_{\bar{r}}} T(q\bar{r}) + h(\bar{r}) \geq \min_{q \in S_{\bar{r}}} \{T(q) + h(\operatorname{Suff}_k(q)) + \hat{\eta}(\operatorname{Suff}_k q,$
 484 $\bar{r})\} = \min_{r' \in \operatorname{Prec}(\bar{r})} \{\hat{W}(r') + \hat{\eta}(r', \bar{r})\} = \hat{W}(\bar{r})$, where we used the
 485 fact that $W(r') = \hat{W}(r')$, that follows from the definition of \bar{r} , since
 486 the value of r' that attains the minimum is such that $W(r') < W(\bar{r})$.
 487 Then, the obtained inequality contradicts the fact that $\hat{W}(\bar{r}) > W(\bar{r})$.
 488 2) Let $A \subset \mathbb{V}$ be the set of values of $r \in \mathbb{V}$ for which 2)
 489 does not hold, and by contradiction, assume that $A \neq \emptyset$ and let
 490 $\hat{r} = \operatorname{argmin}_{r \in A} \hat{W}(r)$. Then, by definition of \hat{r} , it satisfies the
 491 following two properties: $(\forall \bar{r} \in \mathbb{V}_k \mid \hat{W}(\bar{r}) \leq \hat{W}(\hat{r})) \ell^+(\bar{r}) \leq \ell^-(\bar{r})$,
 492 moreover, $\hat{W}(\hat{r}) \neq W(\hat{r})$. Note that, from the definitions of \hat{W} ,
 493 $W(s) = \hat{W}(s)$. Then, $W(\hat{r}) = \min_{p \in P_s \mid \operatorname{Suff}_k p = \hat{r}} T(p) + h(\hat{r}) =$
 494 $\min_{q \in P_s \mid \operatorname{Suff}_k q \in \operatorname{Prec}(\hat{r})} \{T(q\hat{r}) + h(\operatorname{Suff}_k q) - h(\operatorname{Suff}_k q) + h(\hat{r})\} =$
 495 $\min_{r' \in \operatorname{Prec}(\hat{r})} \{\hat{W}(r') + \hat{\eta}(r', \hat{r})\} = \hat{W}(\hat{r})$, which contradicts the
 496 definition of \hat{r} . Here, we used (9) and the fact that, since $\hat{W}(r') < \hat{W}(\hat{r})$
 497 and by the definition of \hat{r} , $\hat{W}(r') = W(r')$. \square

498 Proposition V.4 implies that $\hat{W}(r)$ is a lower bound of $W(r)$ and
 499 that it corresponds to the length of the shortest path from s to r on
 500 the extended directed graph \mathbb{G} , with arc length given in accordance
 501 to (11), namely by the value of function $\hat{\eta}$. Hence, $\hat{W}(f)$ can be
 502 computed by Dijkstra's algorithm (which is equivalent to compute V
 503 with A^* algorithm, with heuristic h). The algorithm that we are going
 504 to present is based on the following basic observation. If A^* algorithm
 505 computes $f^* = \operatorname{argmin}_{f \in \bar{F}} \hat{W}(f)$ by visiting only nodes for which
 506 $\ell^+(r) \leq \ell^-(r)$, then 2) of Proposition V.4 is satisfied for $r = f^*$ and
 507 $\hat{W}(f^*) = W(f^*)$ is the optimal value of the k -BASP. If this is not the
 508 case, we increase k by 1 and rerun the A^* algorithm. Note that the
 509 algorithm starts with $k = 2$, since, according to its definition, $K(\mathbb{B})$
 510 equals 1 only if no acceleration bounds are present and, in this case, the
 511 BASP is equivalent to a standard SP and can be solved by Dijkstra's
 512 algorithm.

513 *Algorithm V.5 (Adaptive A^* algorithm for k -BASP):*

- 514 1) Set $k = 2$.
- 515 2) Execute A^* algorithm, and at every visit of a new node r , if none
 516 of the two conditions $\ell^+(r) \leq \ell^-(r)$ and $|r| < k$ holds, set $k = k + 1$
 517 and repeat step 2).

518 Note that the algorithm does not compute the exact value $K(\mathbb{B})$.
 519 Rather, it underestimates it. More precisely, it stops with the smallest
 520 k value needed to solve the BASP between the given source and
 521 destination nodes. That is, the smallest k that satisfies the k -BASP
 522 definition over the explored subgraph.

523 *Proposition V.6:* Algorithm V.5 terminates with $k \leq K(\mathbb{B})$ and
 524 returns an optimal solution.

525 *Proof:* By Definition (4) of $K(\mathbb{B})$, if $k = K(\mathbb{B})$, the condition
 526 $\ell^+(r) \leq \ell^-(r)$ is satisfied for all r . Hence, there exists $k \leq K(\mathbb{B})$
 527 for which the algorithm terminates. Let $r \in \mathbb{V}_k$, with $\bar{r} \in F$ be the
 528 last-visited node before the termination of the algorithm. By 2) of
 529 Proposition V.4, we have that $\hat{W}(r) = W(r) = V(r)$ (since $h(r) =$
 530 0), but, by definition, $V(r)$ is the shortest time for reaching a node in
 531 F . \square

532 VI. NUMERICAL EXPERIMENTS

533 A. Randomly Generated Problems

534 We performed various tests on problems associated to graphs with n
 535 nodes, for increasing values of n , randomly generated with function `ge-`
 536 `ographical_threshold_graph` of Python package `NetworkX` (`networkx.`
 537 `org`). Essentially, each node is associated to a position randomly chosen
 538 from set $[0, 1]^2$. Edges are randomly determined in such a way that

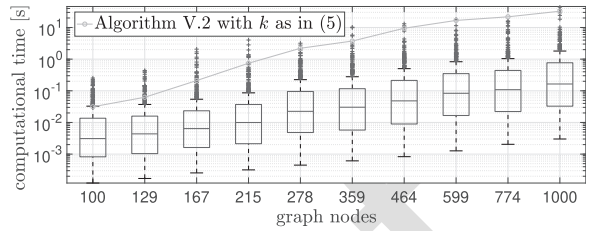


Fig. 5. BASP computing times on graphs of different size.

TABLE I
PERCENTAGES OF k VALUES FOR GRAPHS OF DIFFERENT SIZE

n	$k=3$	$k=4$	$k=5$	$k=6$	\bar{k}	n	$k=3$	$k=4$	$k=5$	$k=6$	\bar{k}
100	80.4%	18.0%	1.6%	0.0%	86	359	61.6%	33.8%	4.4%	0.2%	161
129	81.0%	17.2%	1.8%	0.0%	89	464	60.8%	33.0%	6.0%	0.2%	202
167	77.8%	19.6%	2.0%	0.6%	170	599	51.6%	39.8%	8.2%	0.4%	188
215	72.6%	24.2%	3.2%	0.0%	177	744	49.4%	43.0%	6.4%	1.2%	338
278	63.2%	30.6%	6.2%	0.0%	146	1000	43.6%	46.0%	9.6%	0.8%	300

539 closer nodes have a higher connection probability. We multiplied the
 540 obtained positions by factor $10\sqrt{n}$, in order to obtain the same average
 541 node density independently of n . Then, we associated a random angle θ_i
 542 to each node, obtained from a uniform distribution in $[0, 2\pi]$. In this way,
 543 each node of the random graph is associated to a vehicle configuration,
 544 consisting of a position and an angle. Set $\tau(\theta_i) = [\cos \theta_i, \sin \theta_i]^T$.
 545 Each edge (i, j) is associated to a *Dubins path*, which is defined as the
 546 shortest curve of bounded curvature that connects the configurations
 547 associated to nodes i and j , with initial tangent parallel to $\tau(\theta_i)$ and
 548 final tangent parallel to $\tau(\theta_j)$. We chose the minimum turning radius for
 549 the path associated to edge (i, j) as $r_{ij} = \min\{\ell((i, j)) / (d(\theta_i, \theta_j)), 2\}$
 550 where $d(x, y)$ is the angular distance between angles x and y . We set
 551 the acceleration and deceleration bounds constant for all paths and
 552 equal to 0.1 ms^{-2} . The upper squared speed bound is constant for
 553 each arc and given by $2r$, where r is the minimum curvature radius
 554 of the path associated to the arc. In our tests, we used the adaptive
 555 A^* algorithm (see Algorithm V.5). First, we ran simulations for ten
 556 values of n , logarithmically spaced between 100 and 1000. For each
 557 n , we generated 50 different graphs, and for each one of them, we
 558 ran ten simulations, randomly choosing source and target nodes. Fig. 5
 559 shows the mean values and the distributions of the computational times
 560 of Algorithm V.5 and it also shows the mean computational times of
 561 Algorithm V.2 with k computed as in (5). Note that the mean times of
 562 Algorithm V.2 are at least one order of magnitude higher than those of
 563 Algorithm V.5. Table I shows, for each n , the percentages of k values
 564 returned by Algorithm V.5, and the mean value \bar{k} of k computed as
 565 in (5). Note that the values obtained with (5) are on average 54.8 times
 566 larger than those returned by Algorithm V.5.

567 In Section V, we showed that, for a given problem instance, path p^* ,
 568 corresponding to the solution of the BASP, is in general different from
 569 the path \hat{p} obtained as the solution of the BASP with infinite acceleration
 570 bounds (which, in fact, is an SP) and from the path \tilde{p} obtained as the
 571 solution of SP with edge costs equal to their lengths. We ran some
 572 numerical experiments to compare travel times $T_{\mathbb{B}}(p^*)$ and $T_{\mathbb{B}}(\hat{p})$,
 573 (i.e., the travel time of p^* and the one of \hat{p} on which speed has been
 574 planned using the same acceleration bounds of the BASP), and lengths
 575 $\ell(p^*)$ and $\ell(\tilde{p})$. Namely, we generated 50 different random graphs with
 576 $n = 100$ with the procedure presented previously. For each instance,
 577 we considered ten problems obtained by randomly choosing source and
 578 target nodes. Then, we solved the BASP with different acceleration
 579 bounds α^+ and α^- logarithmically spaced in $[0.01, 1] \text{ ms}^{-2}$, with
 579

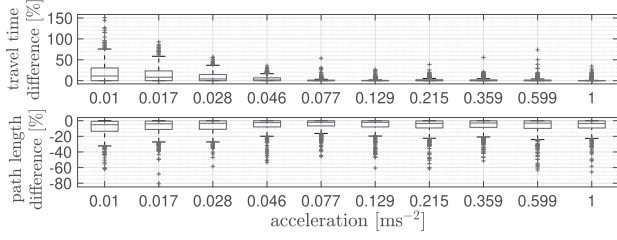


Fig. 6. Travel time difference between BASP and BASP without acceleration bounds and path length difference between BASP and SP with edge costs equal to their lengths.

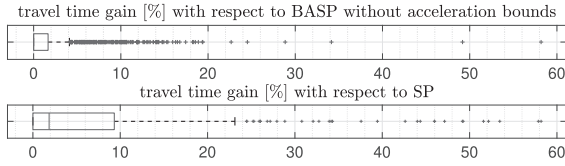


Fig. 7. Travel time gain of BASP on 1000 simulations on the 2 485-node graph with respect to the BASP without acceleration bounds and SP with edge costs equal to their lengths.

580 $\alpha^+ = \alpha^-$. In Fig. 6 (top), we compare the optimal travel times along
 581 paths p^* and \hat{p} , that is, for each value of the acceleration and deceleration
 582 bounds, we report the relative percentage difference $100 \frac{T_{\mathbb{B}}(\hat{p}) - T_{\mathbb{B}}(p^*)}{T_{\mathbb{B}}(p^*)}$
 583 obtained for each test. We observe that for low acceleration and deceler-
 584 ation bounds the difference is more significant, while as the acceleration and deceler-
 585 ation bounds increase, the travel time difference between the
 586 two paths tends to be smaller. This is due to the fact that, if acceleration
 587 bounds are sufficiently high, paths p^* and \hat{p} are the same. In Fig. 6
 588 (bottom), we compare the length of paths p^* and \hat{p} showing how the
 589 BASP solution tends to differ from the SP with edge costs equal to their
 590 lengths even for small acceleration bounds. For p^* and \hat{p} to coincide
 591 one needs even smaller acceleration bounds.

592 B. Real Industrial Applications

593 Here, we present a problem from a real industrial application rep-
 594 resenting an automated warehouse provided by packaging company
 595 Ocme S.r.l., based in Parma, Italy. The problem is described by a graph
 596 of 2 485 nodes and 4 411 arcs. The acceleration and deceleration bounds
 597 are constant, equal for all arcs, and given by $\alpha^+ = 0.28 \text{ ms}^{-2}$ and
 598 $\alpha^- = -0.18 \text{ ms}^{-2}$. The speed bounds are constant for each arc but
 599 vary among different arcs, according to the associated path curvatures,
 600 and they take values on interval $[0.1, 1.7] \text{ ms}^{-1}$. The arc lengths take
 601 values in $[0.2, 18] \text{ m}$ and have an average value of 4.2 m. We ran 1000
 602 simulations by randomly choosing source and the target nodes. The
 603 average value and the standard deviation of the computational time
 604 are 0.1587 and 1.9355 s, respectively. The mean value of k returned
 605 by Algorithm V.5 is 5, while the bound obtained with (5) is 105. We
 606 compare travel times $T_{\mathbb{B}}(p^*)$, $T_{\mathbb{B}}(\hat{p})$, and $T_{\mathbb{B}}(\tilde{p})$, that is, the travel time
 607 of p^* and the ones of \hat{p} and \tilde{p} on which speed has been planned using
 608 the same acceleration bounds of the BASP. Fig. 7 compares the optimal
 609 travel time gain obtained using p^* over \hat{p} and \tilde{p} . Namely, we report
 610 the relative percentage differences over 1000 tests. In the first case, we
 611 had a 2.17% mean gain and the 25% best performing paths p^* had a
 612 8.53% mean gain over \hat{p} . While, in the latter case, we had a 5.85%
 613 mean gain and the 25% best performing paths p^* had a 14.16% mean
 614 gain over \tilde{p} . Note that these results are probably due to the fact that

the graph associated to the industrial problem has a low connectivity. 615
 Indeed, most nodes in the industrial problem represent positions in 616
 corridors and are connected only to the node preceding them and the 617
 one following them along the corridor. Nonetheless, in such industrial 618
 context, even moderate improvements represent a significant gain for a 619
 company. 620

APPENDIX

Proposition A.1: Let $\mu, \alpha : [0, +\infty) \rightarrow \mathbb{R}^+$, for $i \in \{1, 2\}$, let F_i 622
 be the solution of the differential equation (2) where F_i replaces F 623
 and $w_{0,i}$ replaces $\mu(0)$, with $0 \leq w_{0,i} \leq \mu(0)$; and let $\bar{\lambda}$ be such that 624
 $\mu(\bar{\lambda}) = \int_0^{\bar{\lambda}} \alpha(\lambda) d\lambda$. Then, $(\forall \lambda \geq \bar{\lambda}) F_1(\lambda) = F_2(\lambda)$. 625

Proof: Without loss of generality, assume that $w_{0,1} \geq w_{0,2}$. This 626
 implies that $(\forall \lambda \geq 0) F_1(\lambda) \geq F_2(\lambda)$. Indeed, assume by contradic- 627
 tion that there exists $\bar{\lambda}$ such that $F_1(\bar{\lambda}) < F_2(\bar{\lambda})$, then, by conti- 628
 nuity of F_1 and F_2 , this implies that there exists $\hat{\lambda} \leq \bar{\lambda}$ such that 629
 $F_1(\hat{\lambda}) = F_2(\hat{\lambda})$, thus $(\forall \lambda \geq \hat{\lambda}) F_1(\lambda) = F_2(\lambda)$, since, for $\lambda \geq \hat{\lambda}$, 630
 $F_1(\lambda)$ and $F_2(\lambda)$ solve the same differential equation with the same 631
 initial condition at $\lambda = \hat{\lambda}$, contradicting the assumption. Furthermore, 632
 note that $(\exists \bar{\lambda} \in (0, \bar{\lambda}]) F_2(\bar{\lambda}) = \mu(\bar{\lambda})$. Indeed, if by contradiction 633
 $(\forall \lambda \in (0, \bar{\lambda}]) F_2(\lambda) < \mu(\lambda)$, then $(\forall \lambda \in (0, \bar{\lambda}]) F_2'(\lambda) = \alpha(\lambda)$ so that 634
 $F_2(\bar{\lambda}) - F_2(0) = \int_0^{\bar{\lambda}} \alpha(\lambda) d\lambda = \mu(\bar{\lambda})$, which contradicts the assump- 635
 tion. Hence, $(\exists \hat{\lambda} \in \mathbb{R}^+) F_2(\hat{\lambda}) = F_1(\hat{\lambda}) = \mu(\hat{\lambda})$, and consequently, 636
 $(\forall \lambda \geq \hat{\lambda}) F_1(\lambda) = F_2(\lambda)$, which implies the thesis, being $\bar{\lambda} \geq \hat{\lambda}$. \square 637

For $p \in P, \lambda \in [0, \ell(p)]$, we set $\mathcal{W}_p(\lambda) = w$, where w is the solution 638
 of Problem (1) for path p . In other words, $\mathcal{W}_p(\lambda)$ is the square of the 639
 optimal speed profile for traversing the path p , evaluated at arc length 640
 λ , with respect to p . 641

Proposition A.2 1): Let $p_1, p_2, q \in P$, be such that $p_1 q, p_2 q \in P$, 642
 then $(\forall \lambda \geq \ell^+(q)) \mathcal{W}_{p_1 q}(\ell(p_1) + \lambda) = \mathcal{W}_{p_2 q}(\ell(p_2) + \lambda)$. 643

2) Let $p, q_2, q_1 \in P$, be such that $p q_1, p q_2 \in P$, then $(\forall \lambda \leq 644$
 $\ell^-(p)) \mathcal{W}_{p q_1}(\lambda) = \mathcal{W}_{p q_2}(\lambda)$. 645

Proof: We only prove 1), the proof of 2) is analogous. Note 646
 that, for $\lambda \geq 0$, $\mathcal{W}_{p_1 q}(\lambda + \ell(p_1)) = \min\{F_1(\lambda), B(\lambda)\}$, $\mathcal{W}_{p_2 q}(\lambda + 647$
 $\ell(p_2)) = \min\{F_2(\lambda), B(\lambda)\}$, where F_1 and F_2 are the solution of (2) 648
 with $\mu = \mu^+$ and initial conditions $w_{0,1} = \mathcal{W}_{p_1}(\ell(p_1))$ and $w_{0,2} = 649$
 $\mathcal{W}_{p_2}(\ell(p_2))$, respectively, and B is the solution of (3) with $\mu = \mu^+$. 650
 By Proposition A.1, for $\lambda \geq \ell^+(q)$, $F_1(\lambda) = F_2(\lambda)$. Consequently, 651
 $(\forall \lambda \geq \ell^+(q)) \mathcal{W}_{p_1 q}(\ell(p_1) + \lambda) = \mathcal{W}_{p_2 q}(\ell(p_2) + \lambda)$. \square 652

A. Proof of Proposition IV.2

Let Ψ be defined as in (1a), then $T(p_1 t \sigma) - T(p_1 t) = \int_0^{\ell(p_1 t \sigma)} \Psi 654$
 $(\mathcal{W}_{p_1 t \sigma}(\lambda)) d\lambda - \int_0^{\ell(p_1 t)} \Psi(\mathcal{W}_{p_1 t}(\lambda)) d\lambda = \int_{\ell(p_1) + \ell^-(t)}^{\ell(p_1 t \sigma)} \Psi(\mathcal{W}_{p_1 t \sigma}(\lambda)) 655$
 $d\lambda - \int_{\ell(p_1) + \ell^-(t)}^{\ell(p_1 t)} \Psi(\mathcal{W}_{p_1 t}(\lambda)) d\lambda$, where we used that, by 2) of 656
 Proposition A.2, $(\forall \lambda \leq \ell(p_1) + \ell^-(t)) \Psi(\mathcal{W}_{p_1 t \sigma}(\lambda)) = \Psi(\mathcal{W}_{p_1 t}(\lambda))$. 657
 Similarly, $T(p_2 t \sigma) - T(p_2 t) = \int_{\ell(p_2) + \ell^-(t)}^{\ell(p_2 t \sigma)} \Psi(\mathcal{W}_{p_2 t \sigma}(\lambda)) d\lambda - 658$
 $\int_{\ell(p_2) + \ell^-(t)}^{\ell(p_2 t)} \Psi(\mathcal{W}_{p_2 t}(\lambda)) d\lambda$. Moreover, by 1) of Proposition A.2, we 659
 have that $(\forall \lambda \geq \ell^+(t \sigma)) \mathcal{W}_{p_1 t \sigma}(\ell(p_1) + \lambda) d\lambda = \mathcal{W}_{p_2 t \sigma}(\ell(p_2) + \lambda) d\lambda$ 660
 and $(\forall \lambda \geq \ell^+(t)) \mathcal{W}_{p_1 t}(\ell(p_1) + \lambda) d\lambda = \mathcal{W}_{p_2 t}(\ell(p_2) + \lambda) d\lambda$, 661
 which imply that $T(p_1 t \sigma) - T(p_1 t) = T(p_2 t \sigma) - T(p_2 t)$, since 662
 $\ell^+(t) \leq \ell^-(t)$, and as noticed in Section IV, $\ell^+(t \sigma) \leq \ell^+(t)$. \blacksquare 663

REFERENCES

- 664
 [1] A. Abbadi and V. Prenosil, "Safe path planning using cell decomposition 665
 approximation," in *Proc. Int. Conf. Distance Learn. Simul. Commun.*, 666
 May 2015, pp. 8–14. 667
 [2] R. Seif and M. A. Oskoei, "Mobile robot path planning by RRT* in dynamic 668
 environments," *Int. J. Intell. Syst. Appl.*, vol. 7, no. 5, pp. 24–30, 2015. 669

- Q3
- 670 [3] R. V. Cowlagi and P. Tsiotras, "Shortest distance problems in graphs using
671 history-dependent transition costs with application to kinodynamic path
672 planning," in *Proc. Amer. Control Conf.*, 2009, pp. 414–419.
- 673 [4] B. C. Dean, "Shortest paths in FIFO time-dependent networks: Theory and
674 algorithms," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech.
675 Rep., 2004.
- 676 [5] E. He, N. Boland, G. Nehmauser, and M. Savelsbergh, "Time-dependent
677 shortest path problems with penalties and limits on waiting," *INFORMS*
678 *J. Comput.*, vol. 33, no. 3, pp. 997–1014, 2021.
- 679 [6] I. Ioachim, S. Gélinas, F. Soumis, and J. Desrosiers, "A dynamic program-
680 ming algorithm for the shortest path problem with time windows and linear
681 node costs," *Networks*, vol. 31, pp. 193–204, 1998.
- 682 [7] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. New
683 York, NY, USA: McGraw-Hill, 1971.
- 684 [8] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artif.*
685 *Intell.*, vol. 155, no. 1-2, pp. 93–146, 2004.
- 686 [9] D. Ferguson and A. Stentz, "Field D*: An interpolation-based path planner
687 and replanner," *Springer Tracts Adv. Robot.*, vol. 28, pp. 239–253, 2007.
- 688 [10] D. H. Kim, N. T. Hai, and W. Y. Joe, "A guide to selecting path planning
689 algorithm for automated guided vehicle (AGV)," in *AETA 2017—Recent*
690 *Advances in Electrical Engineering and Related Sciences: Theory and*
691 *Application*. Cham, Switzerland: Springer, 2018, pp. 587–596.
- 692 [11] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The
693 path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89,
694 1986.
- [12] M. Raineri, S. Perri, and C. G. L. Bianco, "Online velocity planner for
695 laser guided vehicles subject to safety constraints," in *Proc. IEEE/RSJ Int.*
696 *Conf. Intell. Robots Syst.*, Sep. 2017, pp. 6178–6184. 697
- [13] D. Lam, C. Manzie, M. C. Good, and R. Bitmead, "Receding horizon
698 time-optimal control for a class of differentially flat systems," *Syst. Control*
699 *Lett.*, vol. 83, pp. 61–66, 2015. 700
- [14] T. Faulwasser, V. Hagemeyer, and R. Findeisen, "Optimal exact path-
701 following for constrained differentially flat systems," in *Proc. 18th World*
702 *Congr. Int. Fed. Autom. Control*, 2011, pp. 9875–9880. 703
- [15] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M.
704 Diehl, "Time-optimal path tracking for robots: A convex optimization
705 approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327,
706 Oct. 2009. 707
- [16] L. Consolini, M. Laurini, M. Locatelli, and A. Minari, "A solution of
708 the minimum-time speed planning problem based on lattice theory," *J.*
709 *Franklin Inst.*, vol. 357, pp. 7617–7637, 2020. 710
- [17] S. Ardizzoni, L. Consolini, M. Laurini, and M. Locatelli, "The bounded
711 acceleration shortest path problem: Complexity and solution algo-
712 rithms," 2021, *arXiv:2103.02914*. 713 Q4
- [18] D. Gelperin, "On the optimality of A*," *Artif. Intell.*, vol. 8, no. 1,
714 pp. 69–76, 1977. 715
- [19] S. Edelkamp and S. Schroedl, *Heuristic Search: Theory and Applications*.
716 Elsevier Science, 2011. 717 Q5