

University of Parma Research Repository

Solution algorithms for the Bounded Acceleration Shortest Path Problem

This is the peer reviewd version of the followng 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)

### **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.**
- arked bir anly error provide the state whole proof term state providers. The proof of the state of the st • **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/](https://www.ieee.org/documents/taxonomy_v101.pdf) 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].



 $\mathfrak{p}$ 3

**Q1** 18

# **Technical Notes and Correspondence**

## Solution Algorithms for the Bounded Acceleration Shortest Path Problem

Stefano Ardizzoni<sup>®</sup>, Luca Consolini<sup>®</sup>, Mattia Laurini<sup>®</sup>, and Marco Locatelli<sup>®</sup>

 *Abstract***—The purpose of this article is to introduce and char- acterize 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 acceler- ation 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** 17 **data.**

*Index Terms***—.**

19 **I. INTRODUCTION** 

2<br>
2<br>
Siemano Andrzoni **3** Shortest Path Problem<br>
3<br>
Siemano Andrzoni <sup>3</sup> Dhortest Path Problem<br>
4<br>
Assemblance and the method in the content [of](https://orcid.org/0000-0001-7138-8653) the state of the state of the content of the state of the state of the state 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:* **Q2** *Luca Consolini.)*

The authors are with the Dipartimento di Ingegneria e Architettura, Università degli Studi di Parma, 43124 Parma, Italy (e-mail: [stefano.](mailto:stefano.ardizzoni@unipr.it)<br>ardizzoni@unipr.it: luca.consolini@unipr.it: mattia.laurini@unipr.it: [ardizzoni@unipr.it;](mailto:stefano.ardizzoni@unipr.it) [luca.consolini@unipr.it;](mailto:luca.consolini@unipr.it) [marco.locatelli@unipr.it\)](mailto: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.](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 40 paths  $p_1$  and  $p_2$  connecting positions s and f. The vehicle starts from 41 s with null speed and must reach f with null speed. The solution of SP 42 corresponds to the path  $p_1$ , which is the one of the shortest length. The 43 BASP consists in finding the shortest-time path under acceleration and 44 speed constraints. In this case, we assume that the vehicle acceleration 45 and deceleration are bounded by a common constant and that its speed 46 is bounded only on the central, high-curvature section of  $p_1$ , in order 47 to avoid excessive lateral acceleration, which may cause sideslip. If the 48 bound on acceleration and deceleration is sufficiently high, the solution 49 of the BASP corresponds to the path  $p_2$ . Indeed, even if the latter path is 50 longer, it can be traveled with a greater mean speed. Fig. 1(b) represents 51 the fastest speed profile on  $p_1$ . The x-axis corresponds to the arc-length 52 position on the path  $p_1$  and the y-axis represents the squared speed. 53 In this representation, arc-length intervals of constant acceleration or 54 deceleration correspond to straight lines. Fig. 1(c) represents the fastest 55 speed profile on  $p_2$ . Even if path  $p_2$  is longer than  $p_1$ , it can be traveled 56 in less time. In fact, the vehicle is able to accelerate till the midpoint, 57 and then, to decelerate to the end position  $f$ . 58

The interest for the BASP comes from a specific industrial appli- 59 cation, namely the optimization of automated guided vehicles (AGVs) 60 motion in automated warehouses. The AGVs may be either free to move 61 within a facility or be only allowed to move along predetermined paths. 62 In the first case, one needs to employ environmental representations 63 such as cell decomposition methods [1] or trajectory maps [2]. In par- 64 ticular, the authors in [3] present an algorithm based on a modification of 65 Dijkstra's algorithm in which edge weights are history dependent. Our 66 work is related to the second approach. Namely, we assume that AGVs 67

0018-9286 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

 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 78 planning A<sup>∗</sup> [8], D<sup>∗</sup> [9], and D<sup>∗</sup> 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 104 complexity for fixed values of  $k$ . Since constant  $k$  is problem dependent 105 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.

### 108 **II. NOTATION**

109 A directed graph is a pair  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , where  $\mathbb{V}$  is a set of nodes<br>110 and  $\mathbb{E} \subset \{(x, y) \in \mathbb{V}^2 \mid x \neq y\}$  is a set of directed arcs. A path p on  $\mathbb{G}$ 110 and  $\mathbb{E} \subset \{(x, y) \in \mathbb{V}^2 \mid x \neq y\}$  is a set of directed arcs. A path p on  $\mathbb{G}$ <br>111 is a sequence of adjacent nodes of  $\mathbb{V}$  (i.e.,  $p = \sigma_1 \cdots \sigma_m$ , with  $(\forall i \in$ 111 is a sequence of adjacent nodes of V (i.e.,  $p = \sigma_1 \cdots \sigma_m$ , with  $(\forall i \in \{1, ..., m\})$  ( $\sigma_i, \sigma_{i+1}) \in \mathbb{E}$ ). An alphabet  $\Sigma = {\sigma_1, ..., \sigma_n}$  is a set 112  $\{1,\ldots,m\}$  ( $\sigma_i, \sigma_{i+1} \in \mathbb{E}$ ). An alphabet  $\Sigma = \{\sigma_1,\ldots,\sigma_n\}$  is a set 113 of symbols. A word is any finite sequence of symbols. The set of all of symbols. A word is any finite sequence of symbols. The set of all 114 words over  $\Sigma$  is  $\Sigma^*$ , which also contains the empty word  $\varepsilon$ , while 115  $\Sigma_i$  represents the set of all words of length up to  $i \in \mathbb{N}$ , (i.e., words 115  $\Sigma_i$  represents the set of all words of length up to  $i \in \mathbb{N}$ , (i.e., words<br>116 composed of up to i symbols, including  $\varepsilon$ ). Given a word  $w \in \Sigma^*$ ,  $|w|$ 116 composed of up to i symbols, including  $\varepsilon$ ). Given a word  $w \in \Sigma^*$ ,  $|w|$  denote its length. Given a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , we can think 117 denote its length. Given a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , we can think 118 of  $\mathbb{V}$  as an alphabet so that any path p of  $\mathbb{G}$  is a word in  $\mathbb{V}^*$ . Given of 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 120 of s and t, denoted by  $st \in \Sigma^*$ ; we call t a suffix of st and s a prefix of s and t, denoted by  $st \in \Sigma^*$ ; we call t a suffix of st and s a prefix 121 of st. For  $r \in \mathbb{V}^*$ ,  $\vec{r}$  is the rightmost symbol of r. In the following, we of st. For  $r \in V^*$ ,  $\vec{r}$  is the rightmost symbol of r. In the following, we 122 represent paths of  $G$  as strings of symbols in  $V$ . This allows to use 123 the concatenation operation on paths and to use prefixes and suffixes to 124 represent portions of paths. For  $x \in \mathbb{R}$ ,  $[x] = \min\{i \in \mathbb{Z} \mid i \geq x\}$  is<br>125 the ceiling of x, For  $a, b \in \mathbb{R}$ , we set  $a \wedge b = \min\{a, b\}$  and  $a \vee b =$ 125 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.  $\max\{a, b\}$ , 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 127 space of functions in  $L^{\infty}(I)$  with weak derivative of order 1 with finite 128 space of functions in  $L^{\infty}(I)$  with weak derivative of order 1 with finite 128  $L^{\infty}$ -norm. For  $f, g \in W^{1,\infty}(I)$ , we denote with  $f \wedge g$  and  $f \vee g$  the 129  $\overline{L}^{\infty}$ -norm. For  $f, g \in W^{1,\infty}(I)$ , we denote with  $f \wedge g$  and  $f \vee g$  the 129 point-wise minimum and maximum of  $f$  and  $g$ , respectively. 130 point-wise minimum and maximum of  $f$  and  $g$ , respectively.

### **III. PROBLEM FORMULATION** 131

Before giving the formal description of the BASP, in Section III-A, 132 we briefly discuss the solution of the speed planning problem along a 133 fixed path. Although such problem has been already widely discussed 134 in the literature, here, we briefly describe a way to tackle it in order to 135 better understand the following formulation of the BASP. 136

### *A. Speed Planning Along an Assigned Path* 137

72 metaoro anti traditional internetion based for a strengthenium of the temperature and internetional internetional internetional internet Let  $\gamma : [0, \lambda_f] \to \mathbb{R}^2$  be a  $C^2$  function such that  $(\forall \lambda \in \lambda_{\{f\}})$   $\|\gamma'(\lambda)\| = 1$ . The image set  $\gamma([0, \lambda_f])$  represents the path 139  $[0, \lambda_f]$   $||\gamma'(\lambda)|| = 1$ . The image set  $\gamma([0, \lambda_f])$  represents the path 139<br>followed by a vehicle  $\gamma(0)$  the initial configuration and  $\gamma(\lambda_f)$  the 140 followed by a vehicle,  $\gamma(0)$  the initial configuration, and  $\gamma(\lambda_f)$  the 140 final one. The function  $\gamma$  is an arc-length parameterization of a path. 141 final one. The function  $\gamma$  is an arc-length parameterization of a path. We want to compute the speed law that minimizes the overall travel 142 time while satisfying some kinematic and dynamic constraints. To this 143 end, let  $\xi : [0, t_f] \to [0, \lambda_f]$  be a differentiable monotonically increas-<br>ing function representing the vehicle arc-length coordinate along the 145 ing function representing the vehicle arc-length coordinate along the path as a function of time and let  $v : [0, \lambda_f] \to [0, +\infty)$  be such that 146<br>  $(\forall t \in [0, t_t]) \dot{\xi}(t) = v(\xi(t))$ . In this way,  $v(\lambda)$  is the vehicle speed 147  $(\forall t \in [0, t_f]) \xi(t) = v(\xi(t))$ . In this way,  $v(\lambda)$  is the vehicle speed 147 at position  $\lambda$ . The vehicle position as a function of time is given by 148  $x : [0, t_f] \to \mathbb{R}^2$ ,  $x(t) = \gamma(\xi(t))$ , speed and acceleration are given by 149<br>  $\dot{x}(t) = \gamma'(\xi(t))v(\xi(t))$ , and  $\ddot{x}(t) = a_L(t)\gamma'(\xi(t)) + a_N(t)\gamma'^{\perp}(\xi(t))$ , 150  $\dot{x}(t) = \gamma'(\xi(t))v(\xi(t))$ , and  $\ddot{x}(t) = a_L(t)\gamma'(\xi(t)) + a_N(t)\gamma'^{\perp}(\xi(t))$ , 150<br>where  $a_L(t) = v'(\xi(t))v(\xi(t))$  and  $a_L(t)(t) = v(\xi(t))v(\xi(t))^2$  are 151 where  $a_L(t) = v'(\xi(t))v(\xi(t))$  and  $a_N(t)(t) = \kappa(\xi(t))v(\xi(t))^2$  are 151<br>the longitudinal and normal components of acceleration respectively the longitudinal and normal components of acceleration, respec- 152 tively. Here,  $\kappa : [0, \lambda_f] \to \mathbb{R}$  is the scalar curvature, defined as 153<br> $\kappa(\lambda) = \langle \gamma''(\lambda), \gamma'(\lambda) \rangle$ , where  $\langle \cdot, \cdot \rangle$  denotes the scalar product. 154  $\kappa(\lambda) = \langle \gamma''(\lambda), \gamma'(\lambda)^{\perp} \rangle$ , where  $\langle \cdot, \cdot \rangle$  denotes the scalar product. 154<br>We require to travel distance  $\lambda$ , in a minimum time while satisfy 155

We require to travel distance  $\lambda_f$  in a minimum time while satisfy- 155 ing, for every  $t \in [0, \xi^{-1}(\lambda_f)], 0 \le v^-(\xi(t)) \le v(\xi(t)) \le v^+(\xi(t)),$  156<br>  $|a_N(\xi(t))| \le \beta(\xi(t)), \alpha^-(\xi(t)) \le a_L(\xi(t)) \le \alpha^+(\xi(t)).$  Here, func- $|a_N(\xi(t))| \leq \beta(\xi(t)), \alpha^-(\xi(t)) \leq a_L(\xi(t)) \leq \alpha^+(\xi(t)).$  Here, func-<br>tions  $v^-, v^+, \alpha^-, \alpha^+$ , and  $\beta$  are arc-length-dependent bounds on the tions  $v^-, v^+, \alpha^-, \alpha^+$ , and  $\beta$  are arc-length-dependent bounds on the vehicle speed and on its longitudinal and normal acceleration. It is 159 convenient to make the change of variables  $w = v^2$  (see [15]) so 160<br>that by setting  $\Psi(w) = \int_{0}^{\lambda} f(w(\lambda)) \frac{1}{2} d\lambda w + (\lambda) = v + (\lambda)^2 \wedge \frac{\beta(\lambda)}{\lambda}$  and 161 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 161  $\mu^{-}(\lambda) = v^{-}(\lambda)^{2}$ , our problem takes on the following form. 162

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

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

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

where  $\Psi : W^{1,\infty}([0,\lambda_f]) \to \mathbb{R}$  is order reversing (i.e.,  $(\forall x, y \in \{0, \lambda_f\})$   $x \geq y \Rightarrow \Psi(x) \leq \Psi(y)$ ) and  $\mu^-, \mu^+, \alpha^-, \alpha^+ \in L^{\infty}([0,\lambda_f])$  164  $[0, \lambda_f]$ )  $x \ge y \Rightarrow \Psi(x) \le \Psi(y)$  and  $\mu^-, \mu^+, \alpha^-, \alpha^+ \in L^{\infty}([0, \lambda_f])$  164<br>are assigned functions with  $\mu^-, \alpha^+ > 0$ , and  $\alpha^- < 0$ . Initial and final 165 are assigned functions with  $\mu^-$ ,  $\alpha^+ \ge 0$ , and  $\alpha^- \le 0$ . Initial and final 165<br>conditions on speed can be included in the definition of functions 166 conditions on speed can be included in the definition of functions  $\mu^-$  and  $\mu^+$ . For instance, to set initial condition  $w(0) = w_0$ , it is 167 sufficient to define  $\mu^+(0) = \mu^-(0) = w_0$ . In [16], we introduced a 168 sufficient to define  $\mu^+(0) = \mu^-(0) = w_0$ . In [16], we introduced a 168<br>subset of  $W^{1,\infty}([0,\lambda_f])$ , called Q, as a technical requirement and an 169 subset of  $W^{1,\infty}([0,\lambda_f])$ , called Q, as a technical requirement and an 169 operator based on the solution of the following differential equations: 170 operator based on the solution of the following differential equations:

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

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

171 with  $F, B \in Q$ , that allows to compute the optimal solution of the 172 Problem (1). In particular, in [16], it is shown that the optimal solution is 173  $F(\mu^+) \wedge B(\mu^+)$ . We refer the reader to [16] for a detailed discussion.

### 174 *B. BASP Problem*

T'S which  $P_1 \subset Q$ , but at lower the priori the equinal solution of the BASE Let  $\overline{P_1}(x) = \overline{P_2}(x)$ ,  $\overline{P_3}(x) = \overline{P_3}(x) = \overline{P_4}(x) = \overline{P_5}(x) = \overline{$ 175 In this section, we provide a formal description of the BASP. Let 176 us consider a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , with  $\mathbb{V} = {\sigma_1, \dots, \sigma_N}$ .<br>177 For each  $i \in \{1, \dots, N\}$ , the node  $\sigma_i$  represents an operating point 177 For each  $i \in \{1, ..., N\}$ , the node  $\sigma_i$  represents an operating point 178  $R_i \in \mathbb{R}^2$ . In fact, the restriction  $R_i \in \mathbb{R}^2$  is not strictly necessary but  $R_i \in \mathbb{R}^2$ . In fact, the restriction  $R_i \in \mathbb{R}^2$  is not strictly necessary but 179 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 directed path between two operating points and is associated to an 182 arc-length parameterized path  $\gamma_{\theta}$  of length  $\ell(\theta)$ , such that  $\gamma_{\theta}(0) = R_i$ <br>183 and  $\gamma_{\theta}(\ell(\theta)) = R_i$ . In the following, we denote the set of all possible 183 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<sub>s</sub> the subset paths on G by P. Similarly, for  $s, f \in V$ , we denote by  $P_s$  the subset 185 of P consisting in all paths starting from s and by  $P_{s,f}$  the subset of 186  $P$  consisting in all paths starting from  $s$  and ending in  $f$ . We extend 187 this definition to subsets of V, that is, if  $S, F \subset V$ , we denote by  $P_{S,F}$ 188 the set of all paths starting from nodes in  $S$  and ending in nodes in  $F$ . 189 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,  $\ell(p) = \sum_{n=1}^{m-1} \ell(\sigma_i, \sigma_{i+1})$ . lengths of its individual arcs, that is,  $\ell(p) = \sum_{i=1}^{m-1} \ell(\sigma_i, \sigma_{i+1})$ .<br>To setup our problem, we need to associate four real-valued fun

191 To setup our problem, we need to associate four real-valued functions 192 to each edge  $\theta \in \mathbb{E}$ : the maximum and minimum allowed acceleration 193 and squared speed. The domain of each function is the arc-length 194 coordinate on the path  $\gamma_{\theta}$ . Then, given a specific path p, we are able to 195 define a speed optimization problem of the form (1) by considering the 196 constraints associated to the edges that compose  $p$ . We define the set of edge functions as  $\mathscr{E} = \{ \varphi : \mathbb{E} \times \mathbb{R}^+ \to \mathbb{R} \}$ . If  $\varphi \in \mathscr{E}, \theta \in \mathbb{E}, \lambda \in \mathbb{R}^+$ , 198  $\varphi(\theta, \lambda)$  denotes the value of  $\varphi$  on edge  $\theta$  at position  $\lambda$ . Note that  $\varphi(\theta, \lambda)$ 198  $\varphi(\theta, \lambda)$  denotes the value of  $\varphi$  on edge  $\theta$  at position  $\lambda$ . Note that  $\varphi(\theta, \lambda)$ <br>199 will be relevant only for  $\lambda \in [0, \ell(\theta)]$ . Given a path  $n = \sigma_1 \cdots \sigma_m$ . 199 will be relevant only for  $\lambda \in [0, \ell(\theta)]$ . Given a path  $p = \sigma_1 \cdots \sigma_m$ , 200 we associate to  $\varphi \in \mathscr{E}$  a function  $\varphi_n : [0, \ell(p)] \to \mathbb{R}$  in the following 200 we associate to  $\varphi \in \mathscr{E}$  a function  $\varphi_p : [0, \ell(p)] \to \mathbb{R}$  in the following 201 wav. Define functions  $\Theta : [0, \ell(p)] \to \mathbb{R}$ ,  $\Lambda : [0, \ell(p)] \to \mathbb{R}$  such that 201 way. Define functions  $\Theta : [0, \ell(p)] \to \mathbb{N}$ ,  $\Lambda : [0, \ell(p)] \to \mathbb{R}$  such that<br>202  $\Theta(\lambda) = \max\{i \in \mathbb{N} \mid \ell(\sigma_1 \cdots \sigma_i) \leq \lambda\}$  and  $\Lambda(\lambda) = \ell(\sigma_1 \cdots \sigma_{\Theta(\lambda)})$ . 202  $\Theta(\lambda) = \max\{i \in \mathbb{N} \mid \ell(\sigma_1 \cdots \sigma_i) \leq \lambda\}$  and  $\Lambda(\lambda) = \ell(\sigma_1 \cdots \sigma_{\Theta(\lambda)})$ .<br>203 In this way,  $\Theta(\lambda)$  is such that  $\theta(\lambda) = (\sigma_{\Theta(\lambda)}, \sigma_{\Theta(\lambda)+1})$  is the edge 203 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  $\lambda$  along  $p$ , and  $\Lambda(\lambda)$  is the 204 that contains the position at arc length  $\lambda$  along p, and  $\Lambda(\lambda)$  is the 205 sum of the lengths of all arcs up to node  $\sigma_{\Theta(\lambda)}$  in p. Then, we define sum of the lengths of all arcs up to node  $\sigma_{\Theta(\lambda)}$  in p. Then, we define 206  $\varphi_p(\lambda) = \varphi(\theta(\lambda), \lambda - \Lambda(\lambda)).$ <br>207 Given  $\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^+, \hat{\alpha}^- \in \mathbb{R}$ 

207 Given  $\hat{\mu}^+$ ,  $\hat{\mu}^-$ ,  $\hat{\alpha}^+$ ,  $\hat{\alpha}^- \in \mathscr{E}$  and path  $p \in P$ , let  $\mathbb{B} = (\hat{\mu}^-$ ,  $\hat{\mu}^+$ , 208  $\hat{\alpha}^-$ ,  $\hat{\alpha}^+$ ). Assume  $(\forall \theta \in \mathbb{E})$   $\hat{\mu}^+ (\theta, \cdot) \in Q$  and define  $T_{\mathbb{B}}(p)$ 208  $\hat{\alpha}^-, \hat{\alpha}^+$ ). Assume  $(\forall \theta \in \mathbb{E})$   $\hat{\mu}^+(\theta, \cdot) \in Q$  and define  $T_{\mathbb{B}}(p) =$ <br>209 min<sub>ns</sub> $W^{1,\infty}([0, \infty]) \Psi(w)$ , as the solution of the Problem (1) along 209 min<sub>w∈W</sub><sub>1,∞([0,s<sub>f</sub>])</sub>  $\Psi(w)$ , as the solution of the Problem (1) along 210 path *n* with  $u = \hat{u} - u^+ = \hat{u}^+$ ,  $\alpha^- = \hat{\alpha}^-$ , and  $\alpha^+ = \hat{\alpha}^+$ . In this 210 path p with  $\mu = \hat{\mu}_p^-, \mu^+ = \hat{\mu}_p^+, \alpha^- = \hat{\alpha}_p^-,$  and  $\alpha^+ = \hat{\alpha}_p^+$ . In this 211 way,  $T_{\mathbb{B}}(p)$  is the minimum time required to traverse the path p, 212 respecting the speed and acceleration constraints defined in B. We set respecting the speed and acceleration constraints defined in B. We set 213  $T_{\mathbb{B}}(p) = +\infty$  if the Problem (1) is not feasible.<br>214 The following is the main problem discussed

The following is the main problem discussed in this article.

215 *Problem III.1 (BASP):* Given a graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ ,  $\mu^+, \mu^-,$ 216  $\alpha^-, \alpha^+ \in \mathscr{E}$ ,  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+)$ ,  $s \in \mathbb{V}$ , and  $F \subset \mathbb{V}$ , find 216  $\alpha^-$ ,  $\alpha^+ \in \mathscr{E}$ ,  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+), s \in \mathbb{V}$ , and  $F \subset \mathbb{V}$ , find 217  $\min_{p \in P_s} F \mathbb{B}(p)$ . 217 min<sub>p∈P<sub>s,F</sub> T<sub>B</sub>(p).<br>218 In other words, y</sub>

In other words, we want to find the path  $p$  that minimizes the transfer 219 time between source node  $s$  and a destination node in  $F$ , taking into 220 account bounds on speed and accelerations on each traversed arc (repexample 221 resented by arc functions  $\mu^+, \mu^-, \alpha^-, \alpha^+$ ). The following properties 222 are a direct consequence of the definition of  $T_{\mathbb{B}}(p)$ .<br>223 Proposition III 2: The following properties hold

Proposition III.2: The following properties hold:

224 1) let  $p_1, p_2 \in P$ ,  $p_1p_2 \in P \Rightarrow T_{\mathbb{B}}(p_1p_2) \ge T_{\mathbb{B}}(p_1) + T_{\mathbb{B}}(p_2);$ <br>
225 2) if  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+), \hat{\mathbb{B}} = (\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^-, \hat{\alpha}^+)$  are such the

225 2) if  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+), \mathbb{B} = (\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^-, \hat{\alpha}^+)$  are such that<br>226  $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) [\mu^-(\theta, \lambda), \mu^+(\theta, \lambda)] \subset [\hat{\mu}^-(\theta, \lambda), \hat{\mu}^+(\theta, \lambda)]$  $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) [\mu^-(\theta, \lambda), \mu^+(\theta, \lambda)] \subset [\hat{\mu}^-(\theta, \lambda), \hat{\mu}^+(\theta, \lambda)]$ 



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

and  $[\alpha^-(\theta, \lambda), \alpha^+(\theta, \lambda)] \subset [\hat{\alpha}^-(\theta, \lambda), \hat{\alpha}^+(\theta, \lambda)]$ , then  $(\forall p \in P)$  227<br>  $T_{\mathbb{B}}(p) \ge T_{\hat{\mathbb{B}}}(p)$ . 228  $T_{\mathbb{B}}(p) \geq T_{\hat{\mathbb{B}}}(p)$ . 228<br>In particular, the first property states that the minimum time for 229

traveling the composite path  $p_1p_2$  is greater or equal to the sum of the 230 times needed for traveling  $p_1$  and  $p_2$  separately. In fact, in the first case, 231 the speed must be continuous when passing from  $p_1$  to  $p_2$  (due to the 232 acceleration bounds), but this constraint does not need to be satisfied 233 when the speed profiles for  $p_1$  and  $p_2$  are computed separately. 234

The following proposition (whose proof can be found in [17]) states 235 the theoretical complexity of a simplified version of Problem III.1, 236 called BASP-C, in which maximum and minimum acceleration and 237 speed are constant on each arc. 238

*Proposition III.3:* Problem BASP-C is NP-hard. 239

IV.  $k$ -BASP 240

As we will see in Remark IV.6, SP can be viewed as a special case 241 of the BASP, namely a BASP with unbounded acceleration limits. In 242 fact, also BASP can be viewed as an SP but defined on a different graph 243 with respect to the original one. More precisely, here, we introduce 244 some restrictions on parameters  $\mathbb B$  that allow reducing the BASP to a 245 standard SP that can be solved by Dijkstra's algorithm on an extended 246 graph. Let  $p \in P$ , define 247

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

$$
\ell^-(p) = \max\{\{\lambda \in [0,\ell(p)] \mid -\int_{\lambda}^{\ell(p)} \alpha_p^-(q) dq = \mu_p^+(\lambda)\}, -\infty\}.
$$
249  
In this way  $\ell^+(p)$  is the smallest value of  $\lambda \in [0,\ell(p)]$  for which

In this way,  $\ell^+(p)$  is the smallest value of  $\lambda \in [0, \ell(p)]$  for which 250<br>t solution of F in (2) with  $\alpha^+ = \alpha^+$  starting from initial condithe solution of F in (2), with  $\alpha^+ = \alpha_p^+$ , starting from initial condi-<br>tion  $F(0) = 0$ , reaches the squared speed upper bound  $u^+(1)$ . Note 252 tion  $F(0) = 0$ , reaches the squared speed upper bound  $\mu^+(\lambda)$ . Note 252<br>that  $\ell^+(n) = \infty$  if no such value of  $\lambda$  exists. Similarly,  $\ell^-(n)$  is the 253 that  $\ell^+(p) = \infty$  if no such value of  $\lambda$  exists. Similarly,  $\ell^-(p)$  is the 253<br>largest value of  $\lambda \in [0, \ell(p)]$  for which the solution of B in (3), with 254 largest value of  $\lambda \in [0, \ell(p)]$  for which the solution of B in (3), with 254  $\alpha^- = \alpha^-$  starting from initial condition  $B(\ell(n)) = 0$  reaches  $\mu^+(\lambda) = 255$  $\alpha^- = \alpha_p^-$ , starting from initial condition  $B(\ell(p)) = 0$ , reaches  $\mu^+(\lambda)$ . 255 Again,  $\ell^-(p) = -\infty$  if no such value of  $\lambda$  exists. Note that if  $p, t, pt \in 256$ <br>  $P, \ell^+(pt) < \ell^+(p)$  and  $\ell^-(pt) > \ell^-(p)$  (actually, equalities hold if the 257  $P, \ell^+(pt) \leq \ell^+(p)$  and  $\ell^-(pt) \geq \ell^-(p)$  (actually, equalities hold if the 257 values are all finite). Finally, we define values are all finite). Finally, we define

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

We call k-BASP any instance of Problem III.1 that sat- 259 isfies  $K(\mathbb{B}) \le k$ . For instance, consider the following chain 260<br>graph  $\mathbb{G} = (\mathbb{V} = \{s, 1, 2, f\})$ .  $\mathbb{E} = \{(s, 1), (1, 2), (2, f)\})$ . Here, 261 graph  $\mathbb{G} = (\mathbb{V} = \{s, 1, 2, f\}, \mathbb{E} = \{(s, 1), (1, 2), (2, f)\})$ . Here, 261<br>  $(\forall \theta \in \mathbb{E})$ ,  $\alpha^{-}(\theta) = -1$ ,  $\alpha^{+}(\theta) = 1$ ,  $\mu^{-}(\theta) = 0$ ,  $\ell(\theta) = 1$ , and 262  $(\forall \theta \in \mathbb{E})$   $\alpha^{-}(\theta) = -1$ ,  $\alpha^{+}(\theta) = 1$ ,  $\mu^{-}(\theta) = 0$ ,  $\ell(\theta) = 1$ , and 262<br>  $\mu^{+}((s, 1)) = 1$ ,  $\mu^{+}((1, 2)) = \frac{2}{5}$ ,  $\mu^{+}((2, f)) = 1$ . In this case,  $P_s = 263$  $\mu^+((s, 1)) = 1$ ,  $\mu^+((1, 2)) = \frac{2}{3}$ ,  $\mu^+((2, f)) = 1$ . In this case,  $P_s = 263$ <br>  $\lambda_{s+1} = 12.61$  Moreover  $K(\mathbb{R}) > 2$  since  $\ell^+(*1) = 1 > 0 = 264$  $\{s, s1, s12, s12f\}$ . Moreover,  $K(\mathbb{B}) > 2$ , since  $\ell^+(s1) = 1 > 0 = 264$ <br> $\ell^-(s1)$ , as reported in Fig. 2. Furthermore,  $\ell^+(s12) < \ell^-(s12)$  and 265  $\ell^-(s1)$ , as reported in Fig. 2. Furthermore,  $\ell^+(s12) < \ell^-(s12)$  and 265<br> $\ell^+(12f) < \ell^-(12f)$  and  $s12, 12f$  are the only paths of length 3. Fig. 3 266  $\ell^+(12f) < \ell^-(12f)$  and s12, 12f are the only paths of length 3. Fig. 3 266 shows the computation of  $\ell^+(s12)$  and  $\ell^-(s12)$ ; the computation of 267 shows the computation of  $\ell^+(s12)$  and  $\ell^-(s12)$ ; the computation of 267<br> $\ell^+(12f)$  and  $\ell^-(12f)$  is analogous. Hence, in this example,  $K(\mathbb{B}) = 3$ , 268  $\ell^+(12f)$  and  $\ell^-(12f)$  is analogous. Hence, in this example,  $K(\mathbb{B})=3$ . 268<br>Note that  $K(\mathbb{B})-1$  represents the maximum number of nodes of a 269

Note that  $K(\mathbb{B}) - 1$  represents the maximum number of nodes of a 269 th that can be traveled with a speed profile of maximum acceleration. 270 path that can be traveled with a speed profile of maximum acceleration, followed by one of maximum deceleration, starting and ending with null 271 speed, without violating the maximum speed constraint. The following 272



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

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

$$
K(\mathbb{B}) \leq 1 + \left[2 \max_{\theta \in \mathbb{E}} \frac{\max_{\lambda \in [0,\ell(\theta)]} \mu^+(\theta,\lambda)}{\min_{\lambda \in [0,\ell(\theta)]} (\alpha^+(\theta,\lambda) \wedge |\alpha^-(\theta,\lambda)|) \ell(\theta)}\right].
$$
 (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 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 279 bound (5) overestimates  $K(\mathbb{B})$ . In Section V, we will present a simple method for correctly estimating  $K(\mathbb{B})$ . 280 method for correctly estimating  $K(\mathbb{B})$ .<br>281 We recall that  $\mathbb{V}_b$  represents the subs

IEEE Proof 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  $\varepsilon$ . Define 283 Suff<sub>k</sub>:  $P \to \mathbb{V}_k$  such that, if  $|p| \leq k$ , Suff<sub>k</sub> $(p) = p$  and if  $|p| > k$ , 284 Suff<sub>k</sub> $(p)$  is the suffix of p of length k. The function Suff<sub>k</sub> allows to 284 Suff<sub>k</sub>(p) is the suffix of p of length k. The function Suff<sub>k</sub> allows to introduce a partially defined transition function  $\Gamma : \mathbb{V}_k \times \mathbb{V} \to \mathbb{V}_k$  by 285 introduce a partially defined transition function  $\Gamma : \mathbb{V}_k \times \mathbb{V} \to \mathbb{V}_k$  by 286 setting  $\Gamma(r, \sigma) = \text{Suff}_k(r\sigma)$  if  $r\sigma \in P$ , otherwise, if  $r\sigma \notin P$ ,  $\Gamma(r, \sigma)$ 286 setting  $\Gamma(r, \sigma) = \text{Suff}_k(r\sigma)$  if  $r\sigma \in P$ , otherwise, if  $r\sigma \notin P$ ,  $\Gamma(r, \sigma)$ <br>287 is not defined. Define the incremental cost function  $\eta : P_s \times \mathbb{V} \to \mathbb{R}^+$ 287 is not defined. Define the incremental cost function  $\eta : P_s \times \mathbb{V} \to \mathbb{R}^+$ <br>288 such that, for  $p \in P_s$  and  $q \in \mathbb{V}$ , if  $p q \in P_s$ ,  $p(n, q) = T_{\mathbb{R}}(p q)$ 288 such that, for  $p \in P_s$  and  $\sigma \in \mathbb{V}$ , if  $p\sigma \in P_s$ ,  $\eta(p,\sigma) = T_{\mathbb{B}}(p\sigma) - T_{\mathbb{B}}(p\sigma)$  otherwise  $p(p,\sigma) = +\infty$ . In other words,  $p(p,\sigma)$  is the dif-289  $T_{\mathbb{B}}(p)$ , otherwise  $\eta(p, \sigma) = +\infty$ . In other words,  $\eta(p, \sigma)$  is the dif-<br>290 ference between the minimum time required for traversing p $\sigma$  and the 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:*  $\eta(p, \sigma) \geq T(\sigma)$ .<br>295 *Proof:* By 1) of Proposition III.2.

295 *Proof:* By 1) of Proposition III.2,  $T(p\sigma) \ge T(p) + T(\sigma)$ .  $\Box$ <br>296 The following property, whose proof is presented in the Appendix.

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

*Proposition IV.2:* Let  $p_1, p_2, t \in P$ , if  $p_1t, p_2t \in P$  and  $\ell^+(t) \le$ <br>299  $\ell^-(t)$ , then  $(\forall \sigma \in \mathbb{V}) T(p_1 t \sigma) - T(p_1 t) = T(p_2 t \sigma) - T(p_2 t)$ . 299  $\ell^-(t)$ , then  $(\forall \sigma \in \mathbb{V}) T(p_1 t \sigma) - T(p_1 t) = T(p_2 t \sigma) - T(p_2 t)$ .<br>300 The following is a direct consequence of Proposition IV.2. It

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 Suff<sub>k</sub> (*p*) (its last *k* symbols). 302 depend on the complete path p, but only on Suff  $_k(p)$  (its last k symbols).<br>303 Proposition IV.3: If  $K(\mathbb{B}) < k$  and  $p, p' \in P$  are such that **Proposition IV.3:** If  $K(\mathbb{B}) \le k$  and  $p, p' \in P$  are such that  $\text{Suff}_k(p) = \text{Suff}_k(p')$ , then  $(\forall \sigma \in \mathbb{V}) \eta(p, \sigma) = \eta(p', \sigma)$ . 304 Suff<sub>k</sub> $(p) = \text{Suff}_k(p')$ , then  $(\forall \sigma \in \mathbb{V}) \eta(p, \sigma) = \eta(p, \sigma)$ .<br>305 Define function  $\hat{p} \cdot \mathbb{V}_1 \times \mathbb{V} \to \mathbb{R}^+$  such that  $\hat{p}(r, \sigma)$ .

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

311  $\hat{\eta}$  is the restriction of  $\eta$  on  $\mathbb{V}_k \times \mathbb{V}$ , we denote  $\hat{\eta}$  simply by  $\eta$ .<br>312 The value k can be viewed as the amount of memory rec 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 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 optimal way to complete the path does not only depend on the current 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  $V : \mathbb{V}_k \to \mathbb{R}$  as it. Define function  $V : \mathbb{V}_k \to \mathbb{R}$  as

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



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

Note that the solution of the BASP corresponds to  $\min_{r \in \mathbb{V}_k} |r \in F V(r)$  320 (we recall that  $\vec{r}$  is the last node of r). For  $r \in \mathbb{V}_k$ , define the set of 321 (we recall that  $\vec{r}$  is the last node of r). For  $r \in V_k$ , define the set of predecessors of r as Prec(r) = { $\bar{r} \in V_k | r = \Gamma(\bar{r}, \bar{r})$ . The following 322<br>proposition presents an expression for  $V(r)$  that holds if  $\ell^+(r') \leq 323$ proposition presents an expression for  $V(r)$  that holds if  $\ell^+(r') \leq 323$ <br> $\ell^-(r')$  for all predecessors  $r'$  of r  $\ell^-(r')$  for all predecessors r' of r. ) for all predecessors r' of r.<br>
substitution  $\overline{V}A$ : Let  $r \in \mathbb{N}$ . if  $(\forall r' \in \text{Proc}(r))$   $\ell^+(r') < \ell^-(r')$  225

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

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

*Proof:* Let  $S_r = \{q \in P_s \mid \text{Suff}_k q \vec{r} = r\}$ .  $V(r) = \min_p \in \text{327}$ <br> $|\text{Suff}_k p = rT(p) = \min_{q \in S} \{T(q \vec{r}) - T(q) + T(q)\} = \min_q$  328  $P_s | \text{Suff}_k p = rT(p) = \min_{q \in S_r} \{ T(q\vec{r}) - T(q) + T(q) \} = \min_q$  328<br>  $\in S_r \{ T(q) + T((\text{Suff}_k q)\vec{r}) - T(\text{Suff}_k q) \} = \min_{q \in S_r} \{ T(q) +$  329  $\in S_r\{T(q) + T((\text{Suff}_k q)\vec{r}) - T(\text{Suff}_k q)\} = \min_{q \in S_r}\{T(q) + \eta(r,\vec{r})\} = \min_{q \in S_r}\{\text{Tr}(q) + \eta(r,\vec{r})\} = 330$ 

 $\eta(\text{Suff}_k q, \vec{r})$ } = min<sub>r'</sub>∈Prec(r),  $q \in S_{r'}$ { $T(q) + \eta(r, \vec{r})$ } = 330<br>min<sub>g</sub>'∈Prec(r) { $V(r') + \eta(r, \vec{r})$ }, where we used the facts that 331  $\min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r', \vec{r})\}$ , where we used the facts that 331<br> $T(\alpha \tau) - T(\alpha) - T(\text{Suff}, \alpha \tau) - T(\text{Suff}, \alpha)$  by Proposition IV2  $T(q\sigma) - T(q) = T(\text{Suff}_k q\sigma) - T(\text{Suff}_k q)$ , by Proposition IV.2, 332<br>and that  $q \in P_s$  is such that  $\text{Suff}_k q\vec{r} = r \Leftrightarrow \text{Suff}_k q \in \text{Prec}(r)$ . and that  $q \in P_s$  is such that  $\text{Suff}_k q \vec{r} = r \Leftrightarrow \text{Suff}_k q \in \text{Prec}(r)$ .  $\blacksquare$  333<br>As a consequence of Proposition IV.4. if  $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$ . 334

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

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

*Proposition IV.5:* k-BASP can be solved with complexity 354  $O(|\mathbb{V}|^{k-1}|\mathbb{E}| + (|\mathbb{V}|^k \log |\mathbb{V}|^k)$ <br>Proof: Diikstra's algorithi )). 355<br>m has time complexity  $O(|E| + 356$ 

*Proof:* Dijkstra's algorithm has time complexity  $O(|E| + 356$ <br> $\log |V|$  where  $|E|$  and  $|V|$  are the cardinalities of the edge 257  $|V| \log |V|$ , where  $|E|$  and  $|V|$  are the cardinalities of the edge 357 and vertex sets, respectively. In our case,  $|V| = |\tilde{V}| = |\mathbb{V}_k| = 358$ and vertex sets, respectively. In our case,  $|V| = |\mathbb{V}| = |\mathbb{V}_k| = 358$ <br>  $\sum_{k=1}^{k} |\mathbb{V}|^{i} = O(|\mathbb{V}|^{k})$   $|E| = |\mathbb{V}| \le |\mathbb{V}|$ .  $|\mathbb{F}| = O(|\mathbb{V}|^{k-1} |\mathbb{F}|)$   $\Box$  259  $\sum_{i=0}^{k} |\mathbb{V}|^{i} = O(|\mathbb{V}|^{k}), |E| = |\mathbb{E}| \le |\mathbb{V}_{k-1}\mathbb{E}| = O(|\mathbb{V}|^{k-1}|\mathbb{E}|). \quad \Box$ <br>The following remark establishes that SP can be viewed as a special  $\Box$  359 The following remark establishes that SP can be viewed as a special 360

case of the BASP without acceleration bounds. 361

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

### V. ADAPTIVE A<sup>\*</sup> ALGORITHM FOR *k*-BASP 366

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

369 disadvantages. First,  $\tilde{\mathbb{G}}$  has  $\sum_{j=0}^{k} |\mathbb{V}|^{j}$  nodes so that the time required 370 by Dijkstra's algorithm grows exponentially with  $k$ . We will show that 371 it is possible to mitigate this problem and reduce the number of visited 372 nodes by using the A<sup>∗</sup> algorithm with a suitable heuristic. Second, the 373 estimation of  $k = K(\mathbb{B})$  from its definition is not an easy task. We will 374 show that it is quite easy to adaptively find the correct value of k by show that it is quite easy to adaptively find the correct value of  $k$  by

To enform of  $K = K/3$  from the definition for an expression in this absolute vision in the controlline of Eq. in particular the signal of Eq. in the Sixter Controlline (Eq. in the sixter of Eq. in the Sixter Controlline (F 375 starting from  $k = 2$  and increasing k if needed.<br>376 The A\* algorithm is a heuristic method that The  $A<sup>*</sup>$  algorithm is a heuristic method that allows to compute the 377 optimal path, if it exists (see [18]), by exploring the graph beginning 378 from the starting node along the most promising directions according 379 to a heuristic function that estimates the cost from the current position 380 to the target node. Hence, to implement the  $A^*$  algorithm, we need to 381 define a heuristic function  $h : \mathbb{V}_k \to \mathbb{R}$ , such that, for  $r \in \mathbb{V}_k$ ,  $h(r)$  is a 382 lower bound on  $\min_{p \in P_{-k}} T(p)$ , that is, the minimum time needed for 382 lower bound on  $\min_{p \in P_{\vec{r}, \vec{F}}} T(p)$ , that is, the minimum time needed for 383 traveling from  $\vec{r}$  to a final state in  $\vec{F}$ . In general, we can compute lower 384 bounds for the BASP by relaxing the acceleration constraints  $\alpha^-$  and bounds for the BASP by relaxing the acceleration constraints  $\alpha^-$  and 385  $\alpha^+$ . Namely, let  $\hat{\mathbb{B}}$  be a parameter set obtained by relaxing acceleration 386 constraints in  $\mathbb{B}$ . Then, if  $K(\hat{\mathbb{B}}) < K(\mathbb{B})$ , by Proposition IV.5, the solu-386 constraints in  $\mathbb B$ . Then, if  $K(\hat{\mathbb B}) < K(\mathbb B)$ , by Proposition IV.5, the solu-<br>387 tion of the BASP for parameter  $\hat{\mathbb B}$  can be computed with a lower compution 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 tational time than the solution with parameter  $\mathbb B$ . In particular, we obtain 389 a very simple lower bound by removing acceleration bounds altogether, that is, by setting  $\alpha^- = -\infty$  and  $\alpha^+ = +\infty$ . In this way, the vehicle 391 is allowed to travel at maximum speed everywhere along the path and is allowed to travel at maximum speed everywhere along the path and 392 the incremental cost function  $\eta(p, \sigma)$  is given by the time needed to 393 travel  $\gamma_{\sigma}$  at maximum speed, that is,  $\eta(p, \sigma) = \int_0^{\ell(\vec{p}\sigma)} \frac{1}{\sqrt{\mu^+(\vec{p}, \sigma), \lambda}} d\lambda$ .

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

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

Note that, if  $\alpha^- = -\infty$  and  $\alpha^+ = +\infty$ , h corresponds to the solution of 396 <br>396 1-BASP and all values of h can be efficiently precomputed by Dijkstra's 1-BASP and all values of  $h$  can be efficiently precomputed by Dijkstra's 397 algorithm (see Remark IV.6). The following proposition shows that  $h$ 398 is admissible and consistent so that the  $A^*$  algorithm, with heuristic h,  $399$  provides the optimal solution of the k-BASP and its time complexity 400 is no worse than Dijkstra's algorithm (see [19, Th. 2.9 and 2.10]).

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

402 1) Admissibility:  $(\forall r \in \mathbb{V}_k) h(r) \leq \min_{q \in P_{\vec{r},f}} T_{\mathbb{B}}(q)$ .<br>402 2) Consistency:  $(\forall r \in \mathbb{V}_k) (\forall \sigma \in \mathbb{V}) h(r) \leq n(r,\sigma) +$ 

403 2) Consistency:  $(\forall r \in V_k) (\forall \sigma \in V) h(r) \leq \eta(r, \sigma) + h(\Gamma(r, \sigma)).$ <br>404 Proof: 1)  $h(r) = \min_{\sigma \in P_{r,s}} T_{\mathbb{R}}(p) \leq \min_{\sigma \in P_{r,s}} T_{\mathbb{R}}(q)$ , since  $\hat{\mathbb{B}}$  is 404  $Proof: 1) h(r) = \min_{p \in P_{\vec{r},f}} T_{\hat{\mathbb{B}}}(p) \le \min_{q \in P_{\vec{r},f}} T_{\mathbb{B}}(q)$ , since  $\mathbb{B}$  is 405 a relaxation of B.

406 2)  $h(r) = \min_{p \in P_{\vec{r},f}} T_{\hat{B}}(p) \le T_{\hat{B}}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\hat{B}}(p) \le$ <br>
407  $T_{\infty}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\infty}(\sigma) \le T_{\infty}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\hat{B}}(p) \le$ 407  $T_{\mathbb{B}}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\mathbb{B}}(p) \le \eta(r,\sigma) + \min_{p \in P_{\sigma,f}} T_{\mathbb{B}}(p) =$ <br>408  $\eta(r,\sigma) + h(\Gamma(r,\sigma))$  where  $T_{\mathbb{B}}(\sigma) \le T_{\mathbb{B}}(\sigma)$  by 2) of Proposity 408  $\eta(r,\sigma) + h(\Gamma(r,\sigma))$ , where  $T_{\mathbb{B}}(\sigma) \leq T_{\mathbb{B}}(\sigma)$  by 2) of Proposition III.2<br>409 and  $T_{\mathbb{B}}(\sigma) \leq \eta(r,\sigma)$  by Proposition IV.1.

409 and  $T_{\mathbb{B}}(\sigma) \leq \eta(r, \sigma)$  by Proposition IV.1.  $\Box$ <br>410 Since heuristic h is admissible and consistent, A<sup>\*</sup> is equivalent to  $\Box$ 

411 Dijkstra's algorithm, with the only difference that the incremental cost 412 function  $\eta(r,\sigma)$  is replaced by the modified cost

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

413 (see [19, Lemma 2.3] for a complete discussion). A description of the A<sup>∗</sup> 414 algorithm can be found in literature (for instance, see [19, Algorithm 415 2.13]). We define a priority queue *Q* that contains open nodes, that is, 416 nodes that have already been generated but have not yet been visited. At 7 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 and  $t$  is a lower bound for the time associated to the best completion of 419  $r$  to a path arriving at a final state. We need to perform the following 420 operations on *Q*: operation Insert(*Q*,(*r*, *t*)) inserts couple  $(r, t)$  into<br>421 *Q*: operation  $(r, t) =$  DeleteMin(*Q*) returns the first couple of *Q* 421 *Q*; operation  $(r, t) = \text{DeleteMin}(\mathcal{Q})$  returns the first couple of *Q*, 422 that is, the couple  $(r, t)$  with the minimum time *t*, and removes this 422 that is, the couple  $(r, t)$  with the minimum time t, and removes this couple from  $\mathcal{Q}$ ; and, operation DecreaseKey $(\mathcal{Q}, (r, t))$  assumes that 423 couple from *Q*; and, operation DecreaseKey( $Q$ ,  $(r, t)$ ) assumes that 424  $Q$  already contains a couple  $(r, t')$  with  $t' > t$  and substitutes this 424  $\mathscr Q$  already contains a couple  $(r, t')$  with  $t' > t$  and substitutes this

couple with  $(r, t)$ . Furthermore, we consider three partially defined 425<br>maps value:  $V_L \rightarrow \mathbb{R}$ . parent:  $V_L \rightarrow V_L$ . closed:  $V_L \rightarrow \{0, 1\}$ . 426 maps value :  $\mathbb{V}_k \to \mathbb{R}$ , parent :  $\mathbb{V}_k \to \mathbb{V}_k$ , closed :  $\mathbb{V}_k \to \{0, 1\}$ , 426 such that, for  $r \in \mathbb{V}_k$ , value(*r*) is the current best upper estimate of 427 such that, for  $r \in V_k$ , value $(r)$  is the current best upper estimate of 427  $V(r)$ , parent $(r)$  is the parent node of r, and closed $(r) = 1$  if node 428  $V(r)$ , parent $(r)$  is the parent node of r, and closed $(r)=1$  if node 428<br>r has already been visited. Maps value, parent, and closed can be 429  $r$  has already been visited. Maps value, parent, and closed can be implemented as hash tables. 430

*Algorithm V.2 (A*<sup>∗</sup> *algorithm for* k*-BASP):* 431

1) [initialization] Set  $\mathcal{Q} = \{(s, h(s))\}$ , value $(s) = 0$ . 432<br>2) [expansion] Set  $(r, t) =$  DeleteMin( $\mathcal{Q}$ ) and set closed( $r$ ) = 1. 433

2) [expansion] Set  $(r, t)$  = DeleteMin( $\mathcal{Q}$ ) and set closed( $r$ ) = 1. 433<br> $\vec{r} \in \vec{F}$ , then t is the optimal solution and the algorithm terminates. 434 If  $\vec{r} \in F$ , then t is the optimal solution and the algorithm terminates, 434 returning maps value, parent. Otherwise, for each  $\sigma \in \mathbb{V}$  for which 435 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 closed( $r'$ ) = 436<br>1, go to 3). Else, if  $\text{trall}_0(r')$  is undefined  $\text{Trsort}(\mathcal{Q}(r'/t'))$ . Other 427 1, go to 3). Else, if value(r') is undefined Insert( $\mathcal{Q}, (r, 't')$ ). Oth-<br>erwise if  $t' < \text{value}(r')$  set  $\text{value}(r') = t'$  parent( $r'$ ) = r and do-428 erwise, if  $t' <$  value $(r')$ , set value $(r') = t'$ , parent $(r') = r$  and do 438 DecreaseKey $(\mathscr{Q},(r, 't')).$ )).  $439$ <br>o back to 2) otherwise no solution exists  $440$ 

3) [loop] If  $\mathcal{Q} \neq \emptyset$  go back to 2), otherwise no solution exists. 440<br>Proposition  $V_3$ : Algorithm V2 terminates and returns the optimal 441

*Proposition V.3:* Algorithm V.2 terminates and returns the optimal 441 solution (if it exists), with a time-complexity not higher than Dijkstra's 442 algorithm on the extended graph  $\mathbb{G}$ . 443<br>*Proof*: It is a consequence of the fact that heuristic h is admissible 444

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

Note that, at the end of Algorithm V.2,  $value(f)$  is the optimal value 446 the k-BASP and the optimal path from s to set F can be reconstructed 447 of the  $k$ -BASP and the optimal path from s to set  $F$  can be reconstructed from map parent. <sup>448</sup>

One possible limitation of Algorithm V.2 is that estimating  $K(\mathbb{B})$  449 m its definition can be difficult. A correct estimation of  $K(\mathbb{B})$  is 450 from its definition can be difficult. A correct estimation of  $K(\mathbb{B})$  is 450 critical for the efficiency of the algorithm. Indeed, if  $K(\mathbb{B})$  is overesti-451 critical for the efficiency of the algorithm. Indeed, if  $K(\mathbb{B})$  is overesti-<br>mated, the time complexity of the algorithm is higher than it would be mated, 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, 453<br>Algorithm V.2 is not correct since Proposition IV.4 does not hold. Here, 454 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 455 Algorithm V.2, such that  $k \le K(\mathbb{B})$ , but, in any case, allows to find the 456 optimal solution of the BASP First, we define the modified cost function 457 optimal solution of the BASP. First, we define the modified cost function 457  $W: \mathbb{V}_k \to \mathbb{R}$  as  $W(r) = V(r) + h(r)$ , where V is given by (6) and 458<br>*h* is the heuristic given by (8) If  $(\forall r \in \mathbb{V}_k)$   $\ell^+(r) < \ell^-(r)$  then W is 459 h is the heuristic given by (8). If  $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$ , then W is 459<br>the solution of 460 the solution of

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

Indeed, following the same steps of the proof of Proposition IV.4, 461  $W(r) = V(r) + h(r) = \min_{r' \in \text{Prec}(r)} \{ V(r') + \eta(r', \vec{r}) + h(r) + 462 \}$  $h(r') - h(r')$ } = min<sub>r'</sub>∈Prec(r){ $W(r') + \tilde{\eta}(r', \vec{r})$ }. Hence,  $W(r)$  463 corresponds to the length of the shortest path from s to r on  $\tilde{\mathbb{G}}$ , 464 with arc length given according to  $\tilde{\eta}$ . If condition  $\ell^+(r) \leq \ell^-(r)$  is 465 with arc length given according to  $\tilde{\eta}$ . If condition  $\ell^+(r) \leq \ell^-(r)$  is 465 not satisfied for all  $r \in \mathbb{V}_k$ . (10) does not hold for all  $r \in \mathbb{V}_k$  and 466 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 467 proposition shows that we can still find a lower bound  $\hat{W}$  of W that 468 does correspond to the solution of an SP does correspond to the solution of an SP.

*Proposition V.4:* Let  $\hat{W}: \mathbb{V}_k \to \mathbb{R}$  be the solution of 470

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

where if  $\ell^+(r') \leq \ell^-(r')$  or  $|r'| < k$ ,  $\hat{\eta}(r', \vec{r}) = \tilde{\eta}(r, \vec{r})$ , otherwise 471<br>  $\hat{\eta}(r', \vec{r}) = b(r) - b(r')$ . Then  $(\forall r \in \mathbb{V})$ .  $\hat{\eta}(r', \vec{r}) = h(r) - h(r')$ . Then,  $(\forall r \in \mathbb{V}_k)$  472<br>
1)  $\hat{W}(r) < W(r)$ .

1)  $W(r) \leq W(r)$ ; 473<br>2)  $(\forall \bar{x} \in \mathbb{V}, \pm \hat{W}(\bar{x}) < \hat{W}(r)) \ell^+(\bar{x}) < \ell^-(\bar{x}) \rightarrow \hat{W}(r) = W(r)$  474

2)  $(\forall \bar{r} \in \mathbb{V}_k \mid \hat{W}(\bar{r}) \leq \hat{W}(r)) \ell^+(\bar{r}) \leq \ell^-(\bar{r}) \Rightarrow \hat{W}(r) = W(r).$  474<br>Proof: 1) For  $r \in \mathbb{V}$ . let  $n \in P$  be such that Suff,  $n \in \text{Proc}(r).$  475 *Proof:* 1) For  $r \in V_k$ , let  $p \in P_s$  be such that  $\text{Suff}_k p \in \text{Prec}(r)$ . 475<br>  $\ell^+(\text{Suff}_k p) \leq \ell^-(\text{Suff}_k p)$  in view of Proposition IV2 476 If  $\ell^+$ (Suff<sub>k</sub> p)  $\leq \ell^-$ (Suff<sub>k</sub> p), in view of Proposition IV.2, 476<br> $T(p\vec{r}) = T(p) + n$ (Suff<sub>k</sub> p,  $\vec{r}$ ), otherwise, obviously,  $T(p\vec{r}) \geq T(p)$ , 477  $T(p\vec{r}) = T(p) + \eta(\text{Suff}_k p, \vec{r})$ , otherwise, obviously,  $T(p\vec{r}) \geq T(p)$ . 477<br>Hence, in both cases, by the definition of  $\tilde{\eta}$  in (9),  $T(p\vec{r}) + h(r) > 478$ Hence, in both cases, by the definition of  $\tilde{\eta}$  in (9),  $T(p\tilde{r}) + h(r) \ge 478$ <br>  $T(p) + h(\text{Suff}_k, p) + \hat{\eta}(\text{Suff}_k, p, \tilde{r})$ . By contradiction. assume 479  $T(p) + h(\text{Suff}_k p) + \hat{\eta}(\text{Suff}_k p, \vec{r}).$  By contradiction,

445

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

498 Proposition V.4 implies that  $\hat{W}(r)$  is a lower bound of  $W(r)$  and that it corresponds to the length of the shortest path from s to r on 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 501 to (11), namely by the value of function  $\hat{\eta}$ . Hence,  $\hat{W}(f)$  can be 502 computed by Diikstra's algorithm (which is equivalent to compute V computed by Dijkstra's algorithm (which is equivalent to compute  $V$ 503 with A<sup> $*$ </sup> algorithm, with heuristic h). The algorithm that we are going to present is based on the following basic observation. If A<sup>∗</sup> algorithm 505 computes  $f^* = \argmin_{f \in \tilde{F}} W(f)$  by visiting only nodes for which 506  $f^+(r) < f^-(r)$ , then 2) of Proposition V.4 is satisfied for  $r = f^*$  and  $t^{+}(r) \leq t^{-(r)}(r)$ , then 2) of Proposition V.4 is satisfied for  $r = f^*$  and  $\hat{W}(f^*) = W(f^*)$  is the optimal value of the k-BASP. If this is not the 507  $\dot{W}(f^*) = W(f^*)$  is the optimal value of the k-BASP. If this is not the 508 case we increase k by 1 and rerun the A<sup>\*</sup> algorithm. Note that the 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,  $K(\mathbb{B})$ <br>510 equals 1 only if no acceleration bounds are present and, in this case, the 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.

**algorithm V.5 (Adaptive A<sup>∗</sup> algorithm for k-BASP):** 

514 1) Set  $k = 2$ .<br>515 2) Execute A

2) Execute A<sup> $*$ </sup> 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  $k = k + 1$ <br>517 and repeat step 2). and repeat step 2).

518 Note that the algorithm does not compute the exact value  $K(\mathbb{B})$ .<br>519 Rather, it underestimates it. More precisely, it stops with the smallest 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  $k \le K(\mathbb{B})$  and 524 returns an optimal solution. 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  $k \leq K(\mathbb{B})$ 526  $\ell^+(r) \leq \ell^-(r)$  is satisfied for all r. Hence, there exists  $k \leq K(\mathbb{B})$ <br>527 for which the algorithm terminates. Let  $r \in \mathbb{V}_k$ , with  $\vec{r} \in F$  be the for which the algorithm terminates. Let  $r \in V_k$ , with  $\vec{r} \in F$  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 530 0), but, by definition,  $V(r)$  is the shortest time for reaching a node in 531  $F$ . 531  $F$ .  $\Box$ 

### 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 [\(networkx.](networkx.org) 537 [org\)](networkx.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

	$n   k = 3   k = 4   k = 5   k = 6   k$			$n \mid k = 3 \mid k = 4 \mid k = 5 \mid k = 6 \mid k$		
	$100 80.4\% 18.0\% 1.6\% 0.0\% 86$			$359 \left  61.6\% \right  33.8\% \left  4.4\% \right  0.2\% \left  161 \right $		
	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					

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

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

	travel time gain [%] with respect to BASP without acceleration bounds			
	30 travel time gain $[\%]$ with respect to SP	40	50	60

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.

Fig. 2. These formula methods in the set of t 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 581 paths *p*<sup>∗</sup> and  $\hat{p}$ , that is, for each value of the acceleration and deceleration bounds. we report the relative percentage difference  $100 \frac{T_{\text{B}}(\hat{p}) - T_{\text{B}}(p^*)}{T_{\text{B}}(p^*)}$ bounds, we report the relative percentage difference  $100 \frac{T_B(\hat{p}^*) - T_B(p^*)}{T_B(\hat{p}^*)}$ 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 585 and deceleration bounds increase, the travel time difference between the 586 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<br>588 (bottom), we compare the length of paths  $p^*$  and  $\tilde{p}$  showing how the 588 (bottom), we compare the length of paths  $p^*$  and  $\tilde{p}$  showing how the 588 and pSP solution tends to differ from the SP with edge costs equal to their 589 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. 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  $\alpha^- = -0.18$  ms<sup>-2</sup>. The speed bounds are constant for each arc but vary among different arcs, according to the associated path curvatures. vary among different arcs, according to the associated path curvatures, 600 and they take values on interval  $[0.1, 1.7]$  ms<sup>-1</sup>. The arc lengths take 601 values in  $[0.2, 18]$  m and have an average value of 4.2 m. We ran 1000 601 values in  $[0.2, 18]$  m and have an average value of 4.2 m. We ran 1000 602 simulations by randomly choosing source and the target nodes. The 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 607 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 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 610 the relative percentage differences over 1000 tests. In the first case, we 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% mean gain and the 25% best performing paths  $p^*$  had a 14.16% mean 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 one following them along the corridor. Nonetheless, in such industrial 618 context, even moderate improvements represent a significant gain for a 619

### APPENDIX 621

*Proposition A.1:* Let  $\mu, \alpha : [0, +\infty) \to \mathbb{R}^+$ , for  $i \in \{1, 2\}$ , let  $F_i$  622 the solution of the differential equation (2) where  $F_i$  replaces  $F$  623 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  $\lambda$  be such that 624  $\mu(\bar{\lambda}) = \int_0^{\bar{\lambda}} \alpha(\lambda) d\lambda$ . Then,  $(\forall \lambda \geq \bar{\lambda})$ <br>Proof: Without loss of generali

company. 620

 $\bar{\lambda}$ ) =  $\int_0^{\infty} \alpha(\lambda) d\lambda$ . Then,  $(\forall \lambda \ge \bar{\lambda}) F_1(\lambda) = F_2(\lambda)$ . 625<br>*Proof:* Without loss of generality, assume that  $w_{0,1} \ge w_{0,2}$ . This 626 implies that  $(\forall \lambda \geq 0) F_1(\lambda) \geq F_2(\lambda)$ . Indeed, assume by contradic-<br>tion that there exists  $\overline{\lambda}$  such that  $F_1(\overline{\lambda}) < F_2(\overline{\lambda})$ , then, by conti- 628 tion that there exists  $\lambda$  such that  $F_1(\lambda) < F_2(\lambda)$ , then, by conti- 628<br>puity of E, and E, this implies that there exists  $\hat{\lambda} < \overline{\lambda}$  such that nuity of  $F_1$  and  $F_2$ , this implies that there exists  $\lambda \leq \overline{\lambda}$  such that 629  $F_1(\lambda) = F_2(\lambda)$ , thus  $(\forall \lambda \ge \lambda) F_1(\lambda) = F_2(\lambda)$ , since, for  $\lambda \ge \lambda$ <br> $F_1(\lambda)$  and  $F_2(\lambda)$  solve the same differential equation with the same 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 initial condition at  $\lambda = \lambda$ , contradicting the assumption. Furthermore, 632<br>note that  $(\exists \tilde{\lambda} \in (0, \tilde{\lambda}) \setminus F, (\tilde{\lambda}) = u(\tilde{\lambda})$ . Indeed, if by contradiction 633 note that  $(\exists \lambda \in (0, \lambda]) F_2(\lambda) = \mu(\lambda)$ . Indeed, if by contradiction 633<br>( $\forall \lambda \in (0, \overline{\lambda})$ ) E ( $\lambda$ )  $\leq \mu(\lambda)$  then  $(\forall \lambda \in (0, \overline{\lambda})$ ) E'( $\lambda$ ) =  $\alpha(\lambda)$  so that  $(\forall \lambda \in (0, \bar{\lambda})$   $F_2(\lambda) < \mu(\lambda)$ , 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-<br>time. Hence,  $(\bar{\lambda}) - \bar{F}(\hat{\lambda}) = F(\hat{\lambda})$ , which contradicts the assumption. Hence,  $(\exists \lambda \in \mathbb{R}^+) F_2(\lambda) = F_1(\lambda) = \mu(\lambda)$ , and consequently, 636<br>( $\forall \lambda > \hat{\lambda}$ )  $F_1(\lambda) = F_2(\lambda)$  which implies the thesis being  $\bar{\lambda} > \hat{\lambda}$  $(\forall \lambda \geq \lambda) F_1(\lambda) = F_2(\lambda)$ , which implies the thesis, being  $\overline{\lambda} \geq \lambda$ .  $\square$ <br>For  $n \in P$ ,  $\lambda \in [0, \ell(n)]$ , we set  $\mathcal{W}(1) = uv$ , where w is the solution 637

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

*Proposition A.2 1):* Let  $p_1, p_2, q \in P$ , be such that  $p_1q, p_2q \in P$ , 642 then  $(\forall \lambda \geq \ell^+(q))$   $\mathscr{W}_{p_1q}(\ell(p_1) + \lambda) = \mathscr{W}_{p_2q}(\ell(p_2) + \lambda).$  643<br>2) Let  $p, q_2, q_4 \in P$  be such that  $pq_2, pq_2 \in P$  then  $(\forall \lambda \leq \lambda)$ 

2) Let  $p, q_2, q_1 \in P$ , be such that  $pq_1, pq_2 \in P$ , then  $(\forall \lambda \le 644$  $\ell^-(p)$ )  $\mathscr{W}_{pq_1}(\lambda) = \mathscr{W}_{pq_2}(\lambda)$ . 645<br>*Proof:* We only prove 1), the proof of 2) is analogous. Note 646

that, for  $\lambda \geq 0$ ,  $\mathscr{W}_{p_1q}(\lambda + \ell(p_1)) = \min\{F_1(\lambda), B(\lambda)\}$ ,  $\mathscr{W}_{p_2q}(\lambda + 647)$ <br> $\ell(p_2) = \min\{F_2(\lambda), B(\lambda)\}$ , where  $F_1$  and  $F_2$  are the solution of (2) 648  $\ell(p_2)$  = min{F<sub>2</sub>( $\lambda$ ),  $B(\lambda)$ }, where F<sub>1</sub> and F<sub>2</sub> are the solution of (2) 648<br>with  $\mu = \mu^+$  and initial conditions  $w_0 = \mathcal{W}_+$  ( $\ell(n_1)$ ) and  $w_0 =$  649 with  $\mu = \mu^+$  and initial conditions  $w_{0,1} = \mathcal{W}_{p_1}(\ell(p_1))$  and  $w_{0,2} = 649$ <br> $\mathcal{W}_{\ell}(\ell(p_0))$ , respectively, and B is the solution of (3) with  $\mu = \mu^+$ , 650  $\mathscr{W}_{p_2}(\ell(p_2))$ , respectively, and *B* is the solution of (3) with  $\mu = \mu^+$ . 650 By Proposition A.1, for  $\lambda > \ell^+(q)$ ,  $F_1(\lambda) = F_2(\lambda)$ . Consequently, 651 By Proposition A.1, for  $\lambda \geq \ell^+(q)$ ,  $F_1(\lambda) = F_2(\lambda)$ . Consequently,  $(\forall \lambda > \ell^+(q))$   $\mathcal{W}_{-}(l(n_1) + \lambda) = \mathcal{W}_{-}(l(n_2) + \lambda)$  $(\forall \lambda \geq \ell^+(q))$   $\mathscr{W}_{p_1q}(\ell(p_1) + \lambda) = \mathscr{W}_{p_2q}(\ell(p_2) + \lambda).$ 652

### *A. Proof of Proposition IV.2* 653

Let  $\Psi$  be defined as in (1a), then  $T(p_1 t\sigma) - T(p_1 t) = \int_0^{\ell(p_1 t\sigma)} \Psi$  654  $(\mathscr{W}_{p_1 t\sigma}(\lambda))d\lambda - \int_0^{\ell(p_1t)} \Psi(\mathscr{W}_{p_1t}(\lambda))d\lambda = \int_{\ell(p_1)+\ell^-(t)}^{\ell(p_1t\sigma)} \Psi(\mathscr{W}_{p_1t\sigma}(\lambda))$  655  $d\lambda - \int_{\ell(p_1)}^{\ell(p_1t)} \Psi(\mathscr{W}_{p_1t}(\lambda))d\lambda$ , where we used that, by 2) of 656<br>Preservition 1, (y)  $\ell(\mathscr{W}_{p_1t}(\lambda)) = \ell(\mathscr{W}_{p_2t}(\lambda)) - \frac{1}{2} \mathscr{W}_{p_1t}(\mathscr{W}_{p_2t}(\lambda))$ Proposition A.2,  $(\forall \lambda \leq \ell(p_1) + \ell^-(t)) \Psi(\mathscr{W}_{p_1 t \sigma}(\lambda)) = \Psi(\mathscr{W}_{p_1 t}(\lambda)).$  657 Similarly,  $T(p_2 t\sigma)$  –  $T(p_2 t)$  =  $\int_{\ell(p_2)+\ell^-(t)}^{\ell(p_2 t\sigma)} \Psi(\mathscr{W}_{p_2 t\sigma}(\lambda))d\lambda$  – 658  $\int_{\ell(p_2)+\ell^{-}(t)}^{\ell(p_2t)} \Psi(\mathscr{W}_{p_2t}(\lambda))d\lambda$ . Moreover, by 1) of Proposition A.2, we 659 have that  $(\forall \lambda \geq \ell^+(t\sigma))$   $\mathscr{W}_{p_1t\sigma}(\ell(p_1) + \lambda) d\lambda = \mathscr{W}_{p_2t\sigma}(\ell(p_2) + \lambda) d\lambda$  660<br>and  $(\forall \lambda \geq \ell^+(t))$   $\mathscr{W}_{p_1t}(\ell(p_1) + \lambda) d\lambda = \mathscr{W}_{p_2t}(\ell(p_2) + \lambda) d\lambda$ , 661  $\mathscr{W}_{p_1t}(\ell(p_1) + \lambda)d\lambda = \mathscr{W}_{p_2t}(\ell(p_2) + \lambda)d\lambda$ , 661<br>  $t\sigma$ ) –  $T(n_1t) = T(n_2t\sigma) - T(n_2t)$ , since 662 which imply that  $T(p_1t\sigma) - T(p_1t) = T(p_2t\sigma) - T(p_2t)$ , since 662<br> $\ell^+(t) \leq \ell^-(t)$ , and as noticed in Section IV,  $\ell^+(t\sigma) \leq \ell^+(t)$ .  $\ell^+(t) \leq \ell^-(t)$ , and as noticed in Section IV,  $\ell^+(t\sigma) \leq \ell^+(t)$ .

### 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 andM. A. Oskoei, "Mobile robot path planning by RRT<sup>∗</sup> in dynamic 668 environments," *Int. J. Intell. Syst. Appl.*, vol. 7, no. 5, pp. 24–30, 2015. 669
- 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. 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. **Q3** 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.
	- [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.
	- [8] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A<sup>\*</sup>," Artif. 685 *Intell.*, vol. 155, no. 1-2, pp. 93–146, 2004.
	- 686 [9] D. Ferguson and A. Stentz, "Field D<sup>∗</sup>: An interpolation-based path planner and replanner," *Springer Tracts Adv. Robot.*, vol. 28, pp. 239–253, 2007. 687 and replanner," *Springer Tracts Adv. Robot.*, vol. 28, pp. 239–253, 2007.
	- **For the same of the same of** 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<br>laser quided vehicles subject to safety constraints" in *Proc. IFFF/RS11nt* 696 laser guided vehicles subject to safety constraints," in Proc. IEEE/RSJ Int. *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<br>Lett., vol. 83, pp. 61–66, 2015. *Lett.*, vol. 83, pp. 61–66, 2015.
- [14] T. Faulwasser, V. Hagenmeyer, 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**<br>D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. 704
- [15] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. 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.
- [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 algorithms," 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.
- [19] S. Edelkamp and S. Schroedl, *Heuristic Search: Theory and Applications*. 716 Elsevier Science, 2011.

### **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.**
- artistic lateral procedure is to see the station point and the product of the station o • **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].



 $\overline{2}$ 3

# **Technical Notes and Correspondence**

# Solution Algorithms for the Bounded Acceleration Shortest Path Problem

Stefano Ardizzoni<sup>®</sup>, Luca Consolini<sup>®</sup>, Mattia Laurini<sup>®</sup>, and Marco Locatelli<sup>®</sup>

 *Abstract***—The purpose of this article is to introduce and char- acterize 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 acceler- ation 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** 17 **data.**



**Q1**

18

### 19 I. INTRODUCTION

**Columbian And Contribute Control in the Columbian School and Columbian Sc**  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:* **Q2** *Luca Consolini.)*

The authors are with the Dipartimento di Ingegneria e Architettura, Università degli Studi di Parma, 43124 Parma, Italy (e-mail: stefano.<br>ardizzoni@unipr.it; luca.consolini@unipr.it; mattia.laurini@unipr.it; ardizzoni@unipr.it; luca.consolini@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 40 paths  $p_1$  and  $p_2$  connecting positions s and f. The vehicle starts from 41 s with null speed and must reach  $f$  with null speed. The solution of SP  $\qquad 42$ corresponds to the path  $p_1$ , which is the one of the shortest length. The 43 BASP consists in finding the shortest-time path under acceleration and 44 speed constraints. In this case, we assume that the vehicle acceleration 45 and deceleration are bounded by a common constant and that its speed 46 is bounded only on the central, high-curvature section of  $p_1$ , in order 47 to avoid excessive lateral acceleration, which may cause sideslip. If the 48 bound on acceleration and deceleration is sufficiently high, the solution 49 of the BASP corresponds to the path  $p_2$ . Indeed, even if the latter path is 50 longer, it can be traveled with a greater mean speed. Fig. 1(b) represents 51 the fastest speed profile on  $p_1$ . The x-axis corresponds to the arc-length 52 position on the path  $p_1$  and the y-axis represents the squared speed. 53 In this representation, arc-length intervals of constant acceleration or 54 deceleration correspond to straight lines. Fig. 1(c) represents the fastest 55 speed profile on  $p_2$ . Even if path  $p_2$  is longer than  $p_1$ , it can be traveled 56 in less time. In fact, the vehicle is able to accelerate till the midpoint, 57 and then, to decelerate to the end position  $f$ . 58

The interest for the BASP comes from a specific industrial appli- 59 cation, namely the optimization of automated guided vehicles (AGVs) 60 motion in automated warehouses. The AGVs may be either free to move 61 within a facility or be only allowed to move along predetermined paths. 62 In the first case, one needs to employ environmental representations 63 such as cell decomposition methods [1] or trajectory maps [2]. In par- 64 ticular, the authors in [3] present an algorithm based on a modification of 65 Dijkstra's algorithm in which edge weights are history dependent. Our 66 work is related to the second approach. Namely, we assume that AGVs 67

0018-9286 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

 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 78 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 104 complexity for fixed values of  $k$ . Since constant  $k$  is problem dependent 105 and is not known in advance, in Section V, we present an adaptive  $A^*$ 106 algorithm to find  $k$ . Finally, Section VI presents different computational experiments.

### 108 II. NOTATION

109 A directed graph is a pair  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , where  $\mathbb{V}$  is a set of nodes<br>110 and  $\mathbb{E} \subset \{(x, y) \in \mathbb{V}^2 \mid x \neq y\}$  is a set of directed arcs. A path p on  $\mathbb{G}$ 110 and  $\mathbb{E} \subset \{(x, y) \in \mathbb{V}^2 \mid x \neq y\}$  is a set of directed arcs. A path p on  $\mathbb{G}$ <br>111 is a sequence of adjacent nodes of  $\mathbb{V}$  (i.e.,  $p = \sigma_1 \cdots \sigma_m$ , with  $(\forall i \in$ 111 is a sequence of adjacent nodes of V (i.e.,  $p = \sigma_1 \cdots \sigma_m$ , with  $(\forall i \in \{1, ..., m\})$  ( $\sigma_i, \sigma_{i+1}) \in \mathbb{E}$ ). An alphabet  $\Sigma = {\sigma_1, ..., \sigma_n}$  is a set 112  $\{1,\ldots,m\}$  ( $\sigma_i, \sigma_{i+1} \in \mathbb{E}$ ). An alphabet  $\Sigma = \{\sigma_1,\ldots,\sigma_n\}$  is a set 113 of symbols. A word is any finite sequence of symbols. The set of all of symbols. A word is any finite sequence of symbols. The set of all 114 words over  $\Sigma$  is  $\Sigma^*$ , which also contains the empty word  $\varepsilon$ , while 115  $\Sigma_i$  represents the set of all words of length up to  $i \in \mathbb{N}$ , (i.e., words 115  $\Sigma_i$  represents the set of all words of length up to  $i \in \mathbb{N}$ , (i.e., words<br>116 composed of up to i symbols, including  $\varepsilon$ ). Given a word  $w \in \Sigma^*$ ,  $|w|$ 116 composed of up to i symbols, including  $\varepsilon$ ). Given a word  $w \in \Sigma^*$ ,  $|w|$  denote its length. Given a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , we can think 117 denote its length. Given a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , we can think 118 of  $\mathbb{V}$  as an alphabet so that any path p of  $\mathbb{G}$  is a word in  $\mathbb{V}^*$ . Given of 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 120 of s and t, denoted by  $st \in \Sigma^*$ ; we call t a suffix of st and s a prefix of s and t, denoted by  $st \in \Sigma^*$ ; we call t a suffix of st and s a prefix 121 of st. For  $r \in \mathbb{V}^*$ ,  $\vec{r}$  is the rightmost symbol of r. In the following, we of st. For  $r \in V^*$ ,  $\vec{r}$  is the rightmost symbol of r. In the following, we 122 represent paths of  $G$  as strings of symbols in  $V$ . This allows to use 123 the concatenation operation on paths and to use prefixes and suffixes to 124 represent portions of paths. For  $x \in \mathbb{R}$ ,  $[x] = \min\{i \in \mathbb{Z} \mid i \geq x\}$  is<br>125 the ceiling of x, For  $a, b \in \mathbb{R}$ , we set  $a \wedge b = \min\{a, b\}$  and  $a \vee b =$ 125 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.  $\max\{a, b\}$ , 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 127 space of functions in  $L^{\infty}(I)$  with weak derivative of order 1 with finite 128 space of functions in  $L^{\infty}(I)$  with weak derivative of order 1 with finite 128  $L^{\infty}$ -norm. For  $f, g \in W^{1,\infty}(I)$ , we denote with  $f \wedge g$  and  $f \vee g$  the 129  $\overline{L}^{\infty}$ -norm. For  $f, g \in W^{1,\infty}(I)$ , we denote with  $f \wedge g$  and  $f \vee g$  the 129 point-wise minimum and maximum of  $f$  and  $g$ , respectively. 130 point-wise minimum and maximum of  $f$  and  $g$ , respectively.

### **III. PROBLEM FORMULATION** 131

Before giving the formal description of the BASP, in Section III-A, 132 we briefly discuss the solution of the speed planning problem along a 133 fixed path. Although such problem has been already widely discussed 134 in the literature, here, we briefly describe a way to tackle it in order to 135 better understand the following formulation of the BASP. 136

### *A. Speed Planning Along an Assigned Path* 137

72 metaoro anticonductural technology and internet in the set of th Let  $\gamma : [0, \lambda_f] \to \mathbb{R}^2$  be a  $C^2$  function such that  $(\forall \lambda \in \lambda_i)$   $|\psi'(\lambda)| = 1$ . The image set  $\gamma([0, \lambda_i])$  represents the path 139  $[0, \lambda_f]$   $||\gamma'(\lambda)|| = 1$ . The image set  $\gamma([0, \lambda_f])$  represents the path 139<br>followed by a vehicle  $\gamma(0)$  the initial configuration and  $\gamma(\lambda_f)$  the 140 followed by a vehicle,  $\gamma(0)$  the initial configuration, and  $\gamma(\lambda_f)$  the 140 final one. The function  $\gamma$  is an arc-length parameterization of a path. 141 final one. The function  $\gamma$  is an arc-length parameterization of a path. We want to compute the speed law that minimizes the overall travel 142 time while satisfying some kinematic and dynamic constraints. To this 143 end, let  $\xi$  :  $[0, t_f] \rightarrow [0, \lambda_f]$  be a differentiable monotonically increas-<br>ing function representing the vehicle arc-length coordinate along the 145 ing function representing the vehicle arc-length coordinate along the path as a function of time and let  $v : [0, \lambda_f] \to [0, +\infty)$  be such that 146<br>  $(\forall t \in [0, t] \times \hat{\xi}(t)) = v(\xi(t))$ . In this way,  $v(\lambda)$  is the vehicle speed 147  $(\forall t \in [0, t_f]) \xi(t) = v(\xi(t))$ . In this way,  $v(\lambda)$  is the vehicle speed 147 at position  $\lambda$ . The vehicle position as a function of time is given by 148  $x : [0, t_f] \to \mathbb{R}^2$ ,  $x(t) = \gamma(\xi(t))$ , speed and acceleration are given by 149<br>  $\dot{x}(t) = \gamma'(\xi(t))v(\xi(t))$ , and  $\ddot{x}(t) = a_L(t)\gamma'(\xi(t)) + a_N(t)\gamma'^{\perp}(\xi(t))$ , 150  $\dot{x}(t) = \gamma'(\xi(t))v(\xi(t))$ , and  $\ddot{x}(t) = a_L(t)\gamma'(\xi(t)) + a_N(t)\gamma'^{\perp}(\xi(t))$ , 150<br>where  $a_L(t) = v'(\xi(t))v(\xi(t))$  and  $a_L(t)(t) = v(\xi(t))v(\xi(t))^2$  are 151 where  $a_L(t) = v'(\xi(t))v(\xi(t))$  and  $a_N(t)(t) = \kappa(\xi(t))v(\xi(t))^2$  are 151<br>the longitudinal and normal components of acceleration respectively the longitudinal and normal components of acceleration, respec- 152 tively. Here,  $\kappa : [0, \lambda_f] \to \mathbb{R}$  is the scalar curvature, defined as 153<br> $\kappa(\lambda) = \langle \gamma''(\lambda), \gamma'(\lambda) \rangle$ , where  $\langle \cdot, \cdot \rangle$  denotes the scalar product. 154  $\kappa(\lambda) = \langle \gamma''(\lambda), \gamma'(\lambda)^{\perp} \rangle$ , where  $\langle \cdot, \cdot \rangle$  denotes the scalar product. 154<br>We require to travel distance  $\lambda$ , in a minimum time while satisfy 155

We require to travel distance  $\lambda_f$  in a minimum time while satisfy- 155 ing, for every  $t \in [0, \xi^{-1}(\lambda_f)], 0 \le v^-(\xi(t)) \le v(\xi(t)) \le v^+(\xi(t)),$  156<br> $|a_N(\xi(t))| \le \beta(\xi(t)), \alpha^-(\xi(t)) \le a_L(\xi(t)) \le \alpha^+(\xi(t)).$  Here, func- $|a_N(\xi(t))| \leq \beta(\xi(t)), \alpha^-(\xi(t)) \leq a_L(\xi(t)) \leq \alpha^+(\xi(t)).$  Here, func-<br>tions  $v^-, v^+, \alpha^-, \alpha^+$ , and  $\beta$  are arc-length-dependent bounds on the tions  $v^-, v^+, \alpha^-, \alpha^+$ , and  $\beta$  are arc-length-dependent bounds on the vehicle speed and on its longitudinal and normal acceleration. It is 159 convenient to make the change of variables  $w = v^2$  (see [15]) so 160<br>that by setting  $\Psi(w) = \int_{0}^{\lambda} f(w(\lambda)) \frac{1}{2} d\lambda w + (\lambda) (w(\lambda)) \frac{1}{2} d\lambda w + (\lambda) (w(\lambda)) \frac{1}{2} d\lambda w$  and 161 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 161  $\mu^{-}(\lambda) = v^{-}(\lambda)^{2}$ , our problem takes on the following form. 162

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

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

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

where  $\Psi : W^{1,\infty}([0,\lambda_f]) \to \mathbb{R}$  is order reversing (i.e.,  $(\forall x, y \in \{0, \lambda_f\})$   $x \geq y \Rightarrow \Psi(x) \leq \Psi(y)$ ) and  $\mu^-, \mu^+, \alpha^-, \alpha^+ \in L^{\infty}([0,\lambda_f])$  164  $[0, \lambda_f]$ )  $x \ge y \Rightarrow \Psi(x) \le \Psi(y)$  and  $\mu^-, \mu^+, \alpha^-, \alpha^+ \in L^{\infty}([0, \lambda_f])$  164<br>are assigned functions with  $\mu^-, \alpha^+ > 0$ , and  $\alpha^- < 0$ . Initial and final 165 are assigned functions with  $\mu^-$ ,  $\alpha^+ \ge 0$ , and  $\alpha^- \le 0$ . Initial and final 165<br>conditions on speed can be included in the definition of functions 166 conditions on speed can be included in the definition of functions  $\mu^-$  and  $\mu^+$ . For instance, to set initial condition  $w(0) = w_0$ , it is 167<br>sufficient to define  $\mu^+(0) = \mu^-(0) = w_0$ . In [16], we introduced a 168 sufficient to define  $\mu^+(0) = \mu^-(0) = w_0$ . In [16], we introduced a 168<br>subset of  $W^{1,\infty}([0,\lambda_\epsilon])$ , called *Q*, as a technical requirement and an 169 subset of  $W^{1,\infty}([0,\lambda_f])$ , called  $Q$ , as a technical requirement and an 169 operator based on the solution of the following differential equations: 170 operator based on the solution of the following differential equations:

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

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

171 with  $F, B \in \mathcal{Q}$ , that allows to compute the optimal solution of the 172 Problem (1). In particular, in [16], it is shown that the optimal solution is 173  $F(\mu^+) \wedge B(\mu^+)$ . We refer the reader to [16] for a detailed discussion.

### 174 *B. BASP Problem*

T'S which  $P_1 \subset Q_1$  can allow the prior in compute the equinal solution of the BASE Let  $\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty} \sum_{j=1}^{\infty$ 175 In this section, we provide a formal description of the BASP. Let 176 us consider a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , with  $\mathbb{V} = {\sigma_1, \ldots, \sigma_N}$ .<br>177 For each  $i \in \{1, \ldots, N\}$ , the node  $\sigma_i$  represents an operating point 177 For each  $i \in \{1, ..., N\}$ , the node  $\sigma_i$  represents an operating point 178  $R_i \in \mathbb{R}^2$ . In fact, the restriction  $R_i \in \mathbb{R}^2$  is not strictly necessary but  $R_i \in \mathbb{R}^2$ . In fact, the restriction  $R_i \in \mathbb{R}^2$  is not strictly necessary but 179 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 directed path between two operating points and is associated to an 182 arc-length parameterized path  $\gamma_{\theta}$  of length  $\ell(\theta)$ , such that  $\gamma_{\theta}(0) = R_i$ <br>183 and  $\gamma_{\theta}(\ell(\theta)) = R_i$ . In the following, we denote the set of all possible 183 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<sub>s</sub> the subset paths on G by P. Similarly, for  $s, f \in V$ , we denote by  $P_s$  the subset 185 of P consisting in all paths starting from s and by  $P_{s,f}$  the subset of 186  $P$  consisting in all paths starting from  $s$  and ending in  $f$ . We extend 187 this definition to subsets of V, that is, if  $S, F \subset V$ , we denote by  $P_{S,F}$ 188 the set of all paths starting from nodes in  $S$  and ending in nodes in  $F$ . 189 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,  $\ell(p) = \sum_{n=1}^{m-1} \ell(\sigma_i, \sigma_{i+1})$ . lengths of its individual arcs, that is,  $\ell(p) = \sum_{i=1}^{m-1} \ell(\sigma_i, \sigma_{i+1})$ .<br>1914 To setup our problem, we need to associate four real-valued fun

191 To setup our problem, we need to associate four real-valued functions 192 to each edge  $\theta \in \mathbb{E}$ : the maximum and minimum allowed acceleration 193 and squared speed. The domain of each function is the arc-length 194 coordinate on the path  $\gamma_{\theta}$ . Then, given a specific path p, we are able to 195 define a speed optimization problem of the form (1) by considering the 196 constraints associated to the edges that compose  $p$ . We define the set of edge functions as  $\mathscr{E} = {\varphi : \mathbb{E} \times \mathbb{R}^+ \to \mathbb{R}}$ . If  $\varphi \in \mathscr{E}, \theta \in \mathbb{E}, \lambda \in \mathbb{R}^+$ <br>198  $\varphi(\theta, \lambda)$  denotes the value of  $\varphi$  on edge  $\theta$  at position  $\lambda$ . Note that  $\varphi(\theta, \lambda)$ 198  $\varphi$ (θ, λ) denotes the value of  $\varphi$  on edge θ at position λ. Note that  $\varphi$ (θ, λ) will be relevant only for λ ∈ [0,  $\ell$ (θ)]. Given a path  $p = \sigma_1 \cdots \sigma_m$ . 199 will be relevant only for  $\lambda \in [0, \ell(\theta)]$ . Given a path  $p = \sigma_1 \cdots \sigma_m$ ,<br>200 we associate to  $\varphi \in \mathcal{E}$  a function  $\varphi : [0, \ell(n)] \to \mathbb{R}$  in the following 200 we associate to  $\varphi \in \mathscr{E}$  a function  $\varphi_p : [0, \ell(p)] \to \mathbb{R}$  in the following<br>201 wav. Define functions  $\Theta : [0, \ell(p)] \to \mathbb{R}$ .  $\Lambda : [0, \ell(p)] \to \mathbb{R}$  such that 201 way. Define functions  $\Theta : [0, \ell(p)] \to \mathbb{N}$ ,  $\Lambda : [0, \ell(p)] \to \mathbb{R}$  such that<br>202  $\Theta(\lambda) = \max\{i \in \mathbb{N} \mid \ell(\sigma_1 \cdots \sigma_i) \leq \lambda\}$  and  $\Lambda(\lambda) = \ell(\sigma_1 \cdots \sigma_{\Theta(\lambda)})$ . 202  $\Theta(\lambda) = \max\{i \in \mathbb{N} \mid \ell(\sigma_1 \cdots \sigma_i) \leq \lambda\}$  and  $\Lambda(\lambda) = \ell(\sigma_1 \cdots \sigma_{\Theta(\lambda)})$ .<br>203 In this way,  $\Theta(\lambda)$  is such that  $\theta(\lambda) = (\sigma_{\Theta(\lambda)}, \sigma_{\Theta(\lambda)+1})$  is the edge 203 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  $\lambda$  along  $p$ , and  $\Lambda(\lambda)$  is the 204 that contains the position at arc length  $\lambda$  along p, and  $\Lambda(\lambda)$  is the 205 sum of the lengths of all arcs up to node  $\sigma_{\Theta(\lambda)}$  in p. Then, we define sum of the lengths of all arcs up to node  $\sigma_{\Theta(\lambda)}$  in p. Then, we define 206  $\varphi_p(\lambda) = \varphi(\theta(\lambda), \lambda - \Lambda(\lambda)).$ <br>207 Given  $\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^+, \hat{\alpha}^- \in \mathbb{R}$ 

207 Given  $\hat{\mu}^+$ ,  $\hat{\mu}^-$ ,  $\hat{\alpha}^+$ ,  $\hat{\alpha}^- \in \mathscr{E}$  and path  $p \in P$ , let  $\mathbb{B} = (\hat{\mu}^-$ ,  $\hat{\mu}^+$ , 208  $\hat{\alpha}^-$ ,  $\hat{\alpha}^+$ ). Assume  $(\forall \theta \in \mathbb{E})$   $\hat{\mu}^+ (\theta, \cdot) \in Q$  and define  $T_{\mathbb{B}}(p)$ 208  $\hat{\alpha}^-, \hat{\alpha}^+$ ). Assume  $(\forall \theta \in \mathbb{E})$   $\hat{\mu}^+(\theta, \cdot) \in Q$  and define  $T_{\mathbb{B}}(p) =$ <br>209 min<sub>ns</sub> $W^{1,\infty}([0, \infty]) \Psi(w)$ , as the solution of the Problem (1) along 209 min<sub>w∈W</sub><sub>1,∞([0,s<sub>f</sub>])</sub>  $\Psi(w)$ , as the solution of the Problem (1) along 210 path *n* with  $u = \hat{u} - u^+ = \hat{u}^+$ ,  $\alpha^- = \hat{\alpha}^-$ , and  $\alpha^+ = \hat{\alpha}^+$ . In this 210 path p with  $\mu = \hat{\mu}_p^-, \mu^+ = \hat{\mu}_p^+, \alpha^- = \hat{\alpha}_p^-,$  and  $\alpha^+ = \hat{\alpha}_p^+$ . In this 211 way,  $T_{\mathbb{B}}(p)$  is the minimum time required to traverse the path p, 212 respecting the speed and acceleration constraints defined in B. We set respecting the speed and acceleration constraints defined in B. We set 213  $T_{\mathbb{B}}(p) = +\infty$  if the Problem (1) is not feasible.<br>214 The following is the main problem discussed

The following is the main problem discussed in this article.

215 *Problem III.1 (BASP):* Given a graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ ,  $\mu^+, \mu^-,$ 216  $\alpha^-, \alpha^+ \in \mathscr{E}$ ,  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+)$ ,  $s \in \mathbb{V}$ , and  $F \subset \mathbb{V}$ , find 216  $\alpha^-$ ,  $\alpha^+ \in \mathscr{E}$ ,  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+), s \in \mathbb{V}$ , and  $F \subset \mathbb{V}$ , find 217  $\min_{p \in P_s} F \mathbb{B}(p)$ .

217 min<sub>p∈P<sub>s,F</sub> T<sub>B</sub>(p).<br>218 In other words, y</sub> In other words, we want to find the path  $p$  that minimizes the transfer 219 time between source node  $s$  and a destination node in  $F$ , taking into 220 account bounds on speed and accelerations on each traversed arc (repexample 221 resented by arc functions  $\mu^+, \mu^-, \alpha^-, \alpha^+$ ). The following properties 222 are a direct consequence of the definition of  $T_{\mathbb{B}}(p)$ .<br>223 Proposition III 2: The following properties hold

Proposition III.2: The following properties hold:

224 1) let  $p_1, p_2 \in P$ ,  $p_1p_2 \in P \Rightarrow T_{\mathbb{B}}(p_1p_2) \ge T_{\mathbb{B}}(p_1) + T_{\mathbb{B}}(p_2);$ <br>
225 2) if  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+), \hat{\mathbb{B}} = (\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^-, \hat{\alpha}^+)$  are such the

225 2) if  $\mathbb{B} = (\mu^+, \mu^-, \alpha^-, \alpha^+), \mathbb{B} = (\hat{\mu}^+, \hat{\mu}^-, \hat{\alpha}^-, \hat{\alpha}^+)$  are such that<br>226  $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) [\mu^-(\theta, \lambda), \mu^+(\theta, \lambda)] \subset [\hat{\mu}^-(\theta, \lambda), \hat{\mu}^+(\theta, \lambda)]$  $(\forall \theta \in \mathbb{E}) (\forall \lambda \in [0, \ell(\theta)]) [\mu^-(\theta, \lambda), \mu^+(\theta, \lambda)] \subset [\hat{\mu}^-(\theta, \lambda), \hat{\mu}^+(\theta, \lambda)]$ 



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

and  $[\alpha^-(\theta, \lambda), \alpha^+(\theta, \lambda)] \subset [\hat{\alpha}^-(\theta, \lambda), \hat{\alpha}^+(\theta, \lambda)]$ , then  $(\forall p \in P)$  227<br>  $T_{\mathbb{B}}(p) \ge T_{\hat{\mathbb{B}}}(p)$ . 228  $T_{\mathbb{B}}(p) \geq T_{\hat{\mathbb{B}}}(p)$ . 228<br>In particular, the first property states that the minimum time for 229

traveling the composite path  $p_1p_2$  is greater or equal to the sum of the 230 times needed for traveling  $p_1$  and  $p_2$  separately. In fact, in the first case, 231 the speed must be continuous when passing from  $p_1$  to  $p_2$  (due to the 232 acceleration bounds), but this constraint does not need to be satisfied 233 when the speed profiles for  $p_1$  and  $p_2$  are computed separately. 234

The following proposition (whose proof can be found in [17]) states 235 the theoretical complexity of a simplified version of Problem III.1, 236 called BASP-C, in which maximum and minimum acceleration and 237 speed are constant on each arc. 238

*Proposition III.3:* Problem BASP-C is NP-hard. 239

IV. 
$$
k
$$
-BASP 240

As we will see in Remark IV.6, SP can be viewed as a special case 241 of the BASP, namely a BASP with unbounded acceleration limits. In 242 fact, also BASP can be viewed as an SP but defined on a different graph 243 with respect to the original one. More precisely, here, we introduce 244 some restrictions on parameters  $\mathbb B$  that allow reducing the BASP to a 245 standard SP that can be solved by Dijkstra's algorithm on an extended 246 graph. Let  $p \in P$ , define 247

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

$$
\ell^-(p) = \max\{\{\lambda \in [0,\ell(p)] \mid -\int_{\lambda}^{\ell(p)} \alpha_p^-(q) dq = \mu_p^+(\lambda)\}, -\infty\}.
$$
249  
In this way  $\ell^+(p)$  is the smallest value of  $\lambda \in [0,\ell(p)]$  for which

In this way,  $\ell^+(p)$  is the smallest value of  $\lambda \in [0, \ell(p)]$  for which 250<br>t solution of E in (2) with  $\alpha^+ = \alpha^+$  starting from initial condithe solution of F in (2), with  $\alpha^+ = \alpha_p^+$ , starting from initial condi-<br>tion  $F(0) = 0$ , reaches the squared speed upper bound  $u^+(1)$ . Note 252 tion  $F(0) = 0$ , reaches the squared speed upper bound  $\mu^+(\lambda)$ . Note 252 that  $\ell^+(p) = \infty$  if no such value of  $\lambda$  exists. Similarly,  $\ell^-(p)$  is the 253 that  $\ell^+(p) = \infty$  if no such value of  $\lambda$  exists. Similarly,  $\ell^-(p)$  is the 253<br>largest value of  $\lambda \in [0, \ell(p)]$  for which the solution of B in (3), with 254 largest value of  $\lambda \in [0, \ell(p)]$  for which the solution of B in (3), with 254  $\alpha^- = \alpha^-$  starting from initial condition  $B(\ell(n)) = 0$  reaches  $\mu^+(\lambda) = 255$  $\alpha^- = \alpha_p^-$ , starting from initial condition  $B(\ell(p)) = 0$ , reaches  $\mu^+(\lambda)$ . 255 Again,  $\ell^-(p) = -\infty$  if no such value of  $\lambda$  exists. Note that if  $p, t, pt \in \mathbb{Z}$  256<br>  $P, \ell^+(pt) < \ell^+(p)$  and  $\ell^-(pt) > \ell^-(p)$  (actually, equalities hold if the 257  $P, \ell^+(pt) \leq \ell^+(p)$  and  $\ell^-(pt) \geq \ell^-(p)$  (actually, equalities hold if the 257 values are all finite). Finally, we define values are all finite). Finally, we define

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

We call k-BASP any instance of Problem III.1 that sat- 259 isfies  $K(\mathbb{B}) \le k$ . For instance, consider the following chain 260<br>graph  $\mathbb{G} = (\mathbb{V} = \{s, 1, 2, f\})$ .  $\mathbb{E} = \{(s, 1), (1, 2), (2, f)\})$ . Here, 261 graph  $\mathbb{G} = (\mathbb{V} = \{s, 1, 2, f\}, \mathbb{E} = \{(s, 1), (1, 2), (2, f)\})$ . Here, 261<br>  $(\forall \theta \in \mathbb{E})$ ,  $\alpha^{-}(\theta) = -1$ ,  $\alpha^{+}(\theta) = 1$ ,  $\mu^{-}(\theta) = 0$ ,  $\ell(\theta) = 1$ , and 262  $(\forall \theta \in \mathbb{E})$   $\alpha^{-}(\theta) = -1$ ,  $\alpha^{+}(\theta) = 1$ ,  $\mu^{-}(\theta) = 0$ ,  $\ell(\theta) = 1$ , and 262<br>  $\mu^{+}((s, 1)) = 1$ ,  $\mu^{+}((1, 2)) = \frac{2}{5}$ ,  $\mu^{+}((2, f)) = 1$ . In this case,  $P_s = 263$  $\mu^+((s, 1)) = 1$ ,  $\mu^+((1, 2)) = \frac{2}{3}$ ,  $\mu^+((2, f)) = 1$ . In this case,  $P_s = 263$ <br>  $\lambda_{s+1} = 12 \pm 12 \pm 1$  Moreover  $K(\mathbb{R}) > 2$  since  $\ell^+((s, 1) - 1) > 0 = 264$  $\{s, s1, s12, s12f\}$ . Moreover,  $K(\mathbb{B}) > 2$ , since  $\ell^+(s1) = 1 > 0 = 264$ <br> $\ell^-(s1)$ , as reported in Fig. 2. Furthermore,  $\ell^+(s12) < \ell^-(s12)$  and 265  $\ell^-(s1)$ , as reported in Fig. 2. Furthermore,  $\ell^+(s12) < \ell^-(s12)$  and 265<br> $\ell^+(12f) < \ell^-(12f)$  and s12. 12 f are the only paths of length 3. Fig. 3 266  $\ell^+(12f) < \ell^-(12f)$  and s12, 12f are the only paths of length 3. Fig. 3 266 shows the computation of  $\ell^+(s12)$  and  $\ell^-(s12)$ ; the computation of 267 shows the computation of  $\ell^+(s12)$  and  $\ell^-(s12)$ ; the computation of 267<br> $\ell^+(12f)$  and  $\ell^-(12f)$  is analogous. Hence, in this example,  $K(\mathbb{B}) = 3$ , 268  $\ell^+(12f)$  and  $\ell^-(12f)$  is analogous. Hence, in this example,  $K(\mathbb{B})=3$ . 268<br>Note that  $K(\mathbb{B})=1$  represents the maximum number of nodes of a 269

Note that  $K(\mathbb{B}) - 1$  represents the maximum number of nodes of a 269 th that can be traveled with a speed profile of maximum acceleration. 270 path that can be traveled with a speed profile of maximum acceleration, followed by one of maximum deceleration, starting and ending with null 271 speed, without violating the maximum speed constraint. The following 272



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

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

$$
K(\mathbb{B}) \leq 1 + \left[2 \max_{\theta \in \mathbb{E}} \frac{\max\limits_{\lambda \in [0,\ell(\theta)]} \mu^+(\theta,\lambda)}{\min\limits_{\lambda \in [0,\ell(\theta)]} (\alpha^+(\theta,\lambda) \wedge |\alpha^-(\theta,\lambda)|) \ell(\theta)}\right].
$$
 (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 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 279 bound (5) overestimates  $K(\mathbb{B})$ . In Section V, we will present a simple method for correctly estimating  $K(\mathbb{B})$ . 280 method for correctly estimating  $K(\mathbb{B})$ .<br>281 We recall that  $\mathbb{V}_k$  represents the subs

IF  $\{t^1\}(12) = 1$   $\frac{4}{5} = C^2(12)$   $\lambda$ <br>
Fig. 3. Computation of  $t^1$  (a12) = 1 and  $t$  (a) = 3<br>
Fig. 4. Conclusion of the factorization provides a simple upper bound of  $K(3)$ <br>  $K(2) \le 1 + \left\{\frac{120\lambda_0}{10} \frac{1}{(x^2 + (y^2$ 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  $\varepsilon$ . Define 283 Suff<sub>k</sub>:  $P \to \mathbb{V}_k$  such that, if  $|p| \leq k$ , Suff<sub>k</sub> $(p) = p$  and if  $|p| > k$ , 284 Suff<sub>k</sub> $(p)$  is the suffix of p of length k. The function Suff<sub>k</sub> allows to 284 Suff<sub>k</sub>(p) is the suffix of p of length k. The function Suff<sub>k</sub> allows to introduce a partially defined transition function  $\Gamma : \mathbb{V}_k \times \mathbb{V} \to \mathbb{V}_k$  by 285 introduce a partially defined transition function  $\Gamma : \mathbb{V}_k \times \mathbb{V} \to \mathbb{V}_k$  by 286 setting  $\Gamma(r, \sigma) = \text{Suff}_k(r\sigma)$  if  $r\sigma \in P$ , otherwise, if  $r\sigma \notin P$ ,  $\Gamma(r, \sigma)$ 286 setting  $\Gamma(r, \sigma) = \text{Suff}_k(r\sigma)$  if  $r\sigma \in P$ , otherwise, if  $r\sigma \notin P$ ,  $\Gamma(r, \sigma)$ <br>287 is not defined. Define the incremental cost function  $\eta : P_s \times \mathbb{V} \to \mathbb{R}^+$ 287 is not defined. Define the incremental cost function  $\eta : P_s \times \mathbb{V} \to \mathbb{R}^+$ <br>288 such that, for  $p \in P_s$  and  $q \in \mathbb{V}$ , if  $p q \in P_s$ ,  $n(n, q) = T_{\mathbb{R}}(p q)$ 288 such that, for  $p \in P_s$  and  $\sigma \in \mathbb{V}$ , if  $p\sigma \in P_s$ ,  $\eta(p,\sigma) = T_{\mathbb{B}}(p\sigma) -$ <br>289  $T_{\mathbb{B}}(p)$  otherwise  $p(p,\sigma) = +\infty$ . In other words,  $p(p,\sigma)$  is the dif-289  $T_{\mathbb{B}}(p)$ , otherwise  $\eta(p, \sigma) = +\infty$ . In other words,  $\eta(p, \sigma)$  is the dif-<br>290 ference between the minimum time required for traversing p $\sigma$  and the 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:*  $\eta(p, \sigma) \geq T(\sigma)$ .<br>295 *Proof:* By 1) of Proposition III.2.

295 *Proof:* By 1) of Proposition III.2,  $T(p\sigma) \ge T(p) + T(\sigma)$ .  $\Box$ <br>296 The following property, whose proof is presented in the Appendix.

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

*Proposition IV.2:* Let  $p_1, p_2, t \in P$ , if  $p_1t, p_2t \in P$  and  $\ell^+(t) \le$ <br>299  $\ell^-(t)$ , then  $(\forall \sigma \in \mathbb{V}) T(p_1 t \sigma) - T(p_1 t) = T(p_2 t \sigma) - T(p_2 t)$ . 299  $\ell^-(t)$ , then  $(\forall \sigma \in \mathbb{V}) T(p_1 t \sigma) - T(p_1 t) = T(p_2 t \sigma) - T(p_2 t)$ .<br>300 The following is a direct consequence of Proposition IV.2. It

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). 302 depend on the complete path p, but only on Suff  $_k(p)$  (its last k symbols).<br>303 Proposition IV.3: If  $K(\mathbb{B}) \le k$  and  $p, p' \in P$  are such that **Proposition IV.3:** If  $K(\mathbb{B}) \le k$  and  $p, p' \in P$  are such that  $\text{Suff}_k(p) = \text{Suff}_k(p')$ , then  $(\forall \sigma \in \mathbb{V}) \eta(p, \sigma) = \eta(p', \sigma)$ .

304 Suff<sub>k</sub> $(p) = \text{Suff}_k(p')$ , then  $(\forall \sigma \in \mathbb{V})$   $\eta(p, \sigma) = \eta(p, \sigma)$ . 305 Define function  $\hat{\eta}: \mathbb{V}_k \times \mathbb{V} \to \mathbb{R}^+$ , such that  $\hat{\eta}(r, \sigma) = \eta(p, \sigma)$ <br>306 where  $p \in P$  is any path such that  $r = \text{Suff}_k(p)$ . We set  $\hat{\eta}(r, \sigma) = +\infty$ 306 where  $p \in P$  is any path such that  $r = \text{Suff}_k(p)$ . We set  $\hat{\eta}(r, \sigma) = +\infty$ <br>307 if such path does not exist. Note that the function  $\hat{\eta}$  is well-defined by 307 if such path does not exist. Note that the function  $\hat{\eta}$  is well-defined by 308 Proposition IV.3, being  $n(p, \sigma)$  identical among all paths p such that  $r =$ 308 Proposition IV.3, being  $\eta(p, \sigma)$  identical among all paths p such that  $r = 309$  Suff<sub>k</sub> $(p)$ . In particular, Proposition IV.3 holds for  $p' = \text{Suff}_{k}(p) = r$ Suff<sub>k</sub>(p). In particular, Proposition IV.3 holds for  $p' = \text{Suff}_k(p) = r$ <br>310 so that we can compute  $\hat{n}$  as  $\hat{n}(r, \sigma) = n(r, \sigma)$ . In the following, since 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  $\eta$  on  $\mathbb{V}_k \times \mathbb{V}$ , we denote  $\hat{\eta}$  simply by  $\eta$ . 311  $\hat{\eta}$  is the restriction of  $\eta$  on  $\mathbb{V}_k \times \mathbb{V}$ , we denote  $\hat{\eta}$  simply by  $\eta$ .<br>312 The value k can be viewed as the amount of memory rec

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 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 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  $V : \mathbb{V}_k \to \mathbb{R}$  as

319 it. Define function 
$$
V : \mathbb{V}_k \to \mathbb{R}
$$
 as

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



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

Note that the solution of the BASP corresponds to  $\min_{r \in \mathbb{V}_k} |r \in F V(r)$  320 (we recall that  $\vec{r}$  is the last node of r). For  $r \in \mathbb{V}_k$ , define the set of 321 (we recall that  $\vec{r}$  is the last node of r). For  $r \in V_k$ , define the set of predecessors of r as Prec(r) = { $\bar{r} \in V_k | r = \Gamma(\bar{r}, \bar{r})$ . The following 322<br>proposition presents an expression for  $V(r)$  that holds if  $\ell^+(r') \leq 323$ proposition presents an expression for  $V(r)$  that holds if  $\ell^+(r') \leq 323$ <br> $\ell^-(r')$  for all predecessors  $r'$  of r  $\ell^-(r')$  for all predecessors r' of r. ) for all predecessors r' of r.  $324$ <br>conosition  $\overline{V}A \cdot \overline{I}$  et  $r \in \mathbb{N}$ . if  $(\forall r' \in \text{Proc}(r))$   $\ell^+(r') < \ell^-(r')$ 

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

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

*Proof:* Let  $S_r = \{q \in P_s \mid \text{Suff}_k q \vec{r} = r\}$ .  $V(r) = \min_p \in \text{327}$ <br> $|\text{Suff}_k p = rT(p) = \min_{q \in S} \{T(q \vec{r}) - T(q) + T(q)\} = \min_q$  328  $P_s | \text{Suff}_k p = rT(p) = \min_{q \in S_r} \{ T(q\vec{r}) - T(q) + T(q) \} = \min_q$  328<br>  $\in S_r \{ T(q) + T((\text{Suff}_k q)\vec{r}) - T(\text{Suff}_k q) \} = \min_{q \in S_r} \{ T(q) +$  329  $\in S_r\{T(q) + T((\text{Suff}_k q)\vec{r}) - T(\text{Suff}_k q)\} = \min_{q \in S_r}\{T(q) + \eta(r,\vec{r})\} = \min_{q \in S_r}\{T(q) + \eta(r,\vec{r})\} =$  330

 $\eta(\text{Suff}_k q, \vec{r})$ } = min<sub>r'∈Prec(r)</sub>,  $q \in S_{r'}$ { $T(q) + \eta(r, \vec{r})$ } = 330<br>min<sub>r'∈Prec(r)</sub> { $V(r') + n(r, \vec{r})$ }, where we used the facts that 331  $\min_{r' \in \text{Prec}(r)} \{V(r') + \eta(r', \vec{r})\}$ , where we used the facts that 331<br> $T(\alpha \tau) - T(\alpha) - T(\text{Suff}, \alpha \tau) - T(\text{Suff}, \alpha)$  by Proposition IV2  $T(q\sigma) - T(q) = T(\text{Suff}_k q\sigma) - T(\text{Suff}_k q)$ , by Proposition IV.2, 332<br>and that  $q \in P_s$  is such that  $\text{Suff}_k q\vec{r} = r \Leftrightarrow \text{Suff}_k q \in \text{Prec}(r)$ . and that  $q \in P_s$  is such that  $\text{Suff}_k q \vec{r} = r \Leftrightarrow \text{Suff}_k q \in \text{Prec}(r)$ .  $\blacksquare$  333<br>As a consequence of Proposition IV.4. if  $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$ . 334

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

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

*Proposition IV.5:* k-BASP can be solved with complexity 354  $O(|\mathbb{V}|^{k-1}|\mathbb{E}| + (|\mathbb{V}|^k \log |\mathbb{V}|^k)$ <br>Proof: Diikstra's algorithi )). 355<br>m has time complexity  $O(|E| + 356$ 

*Proof:* Dijkstra's algorithm has time complexity  $O(|E| + 356$ <br> $\log |V|$  where  $|E|$  and  $|V|$  are the cardinalities of the edge 257  $|V| \log |V|$ , where  $|E|$  and  $|V|$  are the cardinalities of the edge 357 and vertex sets, respectively. In our case,  $|V| = |\tilde{V}| = |\mathbb{V}_k| = 358$ and vertex sets, respectively. In our case,  $|V| = |\mathbb{V}| = |\mathbb{V}_k| = 358$ <br>  $\sum_{k=1}^{k} |\mathbb{V}|^{i} = O(|\mathbb{V}|^{k})$   $|E| = |\mathbb{V}| \le |\mathbb{V}|$ .  $|\mathbb{F}| = O(|\mathbb{V}|^{k-1} |\mathbb{F}|)$   $\Box$  259  $\sum_{i=0}^{k} |\mathbb{V}|^{i} = O(|\mathbb{V}|^{k}), |E| = |\mathbb{E}| \le |\mathbb{V}_{k-1}\mathbb{E}| = O(|\mathbb{V}|^{k-1}|\mathbb{E}|). \quad \Box$ <br>The following remark establishes that SP can be viewed as a special  $\Box$  359

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

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

### V. ADAPTIVE A<sup>∗</sup> ALGORITHM FOR k-BASP 366

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

369 disadvantages. First,  $\tilde{\mathbb{G}}$  has  $\sum_{j=0}^{k} |\mathbb{V}|^j$  nodes so that the time required 370 by Dijkstra's algorithm grows exponentially with  $k$ . We will show that 371 it is possible to mitigate this problem and reduce the number of visited 372 nodes by using the A<sup>∗</sup> algorithm with a suitable heuristic. Second, the 373 estimation of  $k = K(\mathbb{B})$  from its definition is not an easy task. We will 374 show that it is quite easy to adaptively find the correct value of k by show that it is quite easy to adaptively find the correct value of  $k$  by

To enform of  $K = K/3$  from the definition for an expression in this absolution vision in the signature of  $K/3$  in the signature vision for  $K/3$  from the signature of  $K/3$  in the signature of  $K/3$  in the signature of  $K$ 375 starting from  $k = 2$  and increasing k if needed.<br>376 The A<sup>\*</sup> algorithm is a heuristic method that The  $A^*$  algorithm is a heuristic method that allows to compute the 377 optimal path, if it exists (see [18]), by exploring the graph beginning 378 from the starting node along the most promising directions according 379 to a heuristic function that estimates the cost from the current position 380 to the target node. Hence, to implement the  $A^*$  algorithm, we need to 381 define a heuristic function  $h : \mathbb{V}_k \to \mathbb{R}$ , such that, for  $r \in \mathbb{V}_k$ ,  $h(r)$  is a 382 lower bound on  $\min_{p \in P_{-k}} T(p)$ , that is, the minimum time needed for 382 lower bound on  $\min_{p \in P_{\vec{r}, \vec{F}}} T(p)$ , that is, the minimum time needed for 383 traveling from  $\vec{r}$  to a final state in  $\vec{F}$ . In general, we can compute lower 384 bounds for the BASP by relaxing the acceleration constraints  $\alpha^-$  and bounds for the BASP by relaxing the acceleration constraints  $\alpha^-$  and 385  $\alpha^+$ . Namely, let  $\hat{\mathbb{B}}$  be a parameter set obtained by relaxing acceleration 386 constraints in  $\mathbb{B}$ . Then, if  $K(\hat{\mathbb{B}}) < K(\mathbb{B})$ , by Proposition IV.5, the solu-386 constraints in  $\mathbb B$ . Then, if  $K(\hat{\mathbb B}) < K(\mathbb B)$ , by Proposition IV.5, the solu-<br>387 tion of the BASP for parameter  $\hat{\mathbb B}$  can be computed with a lower compution 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 tational time than the solution with parameter  $\mathbb B$ . In particular, we obtain 389 a very simple lower bound by removing acceleration bounds altogether, that is, by setting  $\alpha^- = -\infty$  and  $\alpha^+ = +\infty$ . In this way, the vehicle 391 is allowed to travel at maximum speed everywhere along the path and is allowed to travel at maximum speed everywhere along the path and 392 the incremental cost function  $\eta(p, \sigma)$  is given by the time needed to 393 travel  $\gamma_{\sigma}$  at maximum speed, that is,  $\eta(p, \sigma) = \int_0^{\ell(\vec{p}\sigma)} \frac{1}{\sqrt{\mu^+(\vec{p}, \sigma), \lambda}} d\lambda$ .

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

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

Note that, if  $\alpha^- = -\infty$  and  $\alpha^+ = +\infty$ , h corresponds to the solution of 396 <br>396 1-BASP and all values of h can be efficiently precomputed by Dijkstra's 1-BASP and all values of  $h$  can be efficiently precomputed by Dijkstra's 397 algorithm (see Remark IV.6). The following proposition shows that  $h$ 398 is admissible and consistent so that the  $A^*$  algorithm, with heuristic h,  $399$  provides the optimal solution of the k-BASP and its time complexity 400 is no worse than Dijkstra's algorithm (see [19, Th. 2.9 and 2.10]).

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

402 1) Admissibility:  $(\forall r \in \mathbb{V}_k) h(r) \le \min_{q \in P_{\vec{r},f}} T_{\mathbb{B}}(q)$ .<br>402 2) Consistency:  $(\forall r \in \mathbb{V}_k) (\forall \sigma \in \mathbb{V}) h(r) \le n(r, \sigma) +$ 

403 2) Consistency:  $(\forall r \in V_k) (\forall \sigma \in V) h(r) \leq \eta(r, \sigma) + h(\Gamma(r, \sigma)).$ <br>404 Proof: 1)  $h(r) = \min_{\sigma \in P_{r,s}} T_{\mathbb{R}}(p) \leq \min_{\sigma \in P_{r,s}} T_{\mathbb{R}}(q)$ , since  $\hat{\mathbb{B}}$  is 404  $Proof: 1) h(r) = \min_{p \in P_{\vec{r},f}} T_{\hat{\mathbb{B}}}(p) \le \min_{q \in P_{\vec{r},f}} T_{\mathbb{B}}(q)$ , since  $\mathbb{B}$  is 405 a relaxation of B.

406  $h(r) = \min_{p \in P_{\vec{r},f}} T_{\hat{\mathbb{B}}}(p) \le T_{\hat{\mathbb{B}}}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\hat{\mathbb{B}}}(p) \le$ <br>407  $T_{\mathbb{B}}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\hat{\mathbb{B}}}(p) \le T_{\hat{\mathbb{B}}}(p) + \min_{p \in P_{\sigma,f}} T_{\hat{\mathbb{B}}}(p)$ 407  $T_{\mathbb{B}}(\sigma) + \min_{p \in P_{\sigma,f}} T_{\mathbb{B}}(p) \le \eta(r,\sigma) + \min_{p \in P_{\sigma,f}} T_{\mathbb{B}}(p) =$ <br>409  $n(r,\sigma) + h(\Gamma(r,\sigma))$  where  $T_{\mathbb{B}}(\sigma) \le T_{\mathbb{B}}(\sigma)$  by 2) of Proposity

408  $\eta(r,\sigma) + h(\Gamma(r,\sigma))$ , where  $T_{\mathbb{B}}(\sigma) \leq T_{\mathbb{B}}(\sigma)$  by 2) of Proposition III.2<br>409 and  $T_{\mathbb{B}}(\sigma) \leq \eta(r,\sigma)$  by Proposition IV.1. 409 and  $T_{\mathbb{B}}(\sigma) \leq \eta(r,\sigma)$  by Proposition IV.1.<br>410 Since heuristic *h* is admissible and con  $\Box$ 

Since heuristic  $h$  is admissible and consistent,  $A^*$  is equivalent to 411 Dijkstra's algorithm, with the only difference that the incremental cost 412 function  $\eta(r,\sigma)$  is replaced by the modified cost

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

413 (see [19, Lemma 2.3] for a complete discussion). A description of the A<sup>∗</sup> 414 algorithm can be found in literature (for instance, see [19, Algorithm 415 2.13]). We define a priority queue *Q* that contains open nodes, that is, 416 nodes that have already been generated but have not yet been visited. At  $\sum_{k=1}^{\infty}$  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 and  $t$  is a lower bound for the time associated to the best completion of 419  $r$  to a path arriving at a final state. We need to perform the following 420 operations on *Q*: operation Insert $(\mathcal{Q}, (r, t))$  inserts couple  $(r, t)$  into<br>421  $\mathcal{Q}$ : operation  $(r, t)$  = DeleteMin( $\mathcal{Q}$ ) returns the first couple of  $\mathcal{Q}$ 421 *Q*; operation  $(r, t) = \text{DeleteMin}(\mathcal{Q})$  returns the first couple of *Q*, 422 that is, the couple  $(r, t)$  with the minimum time *t*, and removes this 422 that is, the couple  $(r, t)$  with the minimum time t, and removes this couple from  $\mathcal{Q}$ ; and, operation DecreaseKey $(\mathcal{Q}, (r, t))$  assumes that 423 couple from *Q*; and, operation DecreaseKey( $Q$ ,  $(r, t)$ ) assumes that 424  $Q$  already contains a couple  $(r, t')$  with  $t' > t$  and substitutes this 424  $\mathscr Q$  already contains a couple  $(r, t')$  with  $t' > t$  and substitutes this

couple with  $(r, t)$ . Furthermore, we consider three partially defined 425 maps value:  $V_L \rightarrow \mathbb{R}$ . parent:  $V_L \rightarrow V_L$ . closed:  $V_L \rightarrow \{0, 1\}$ . 426 maps value :  $\mathbb{V}_k \to \mathbb{R}$ , parent :  $\mathbb{V}_k \to \mathbb{V}_k$ , closed :  $\mathbb{V}_k \to \{0, 1\}$ , 426 such that, for  $r \in \mathbb{V}_k$ , value(*r*) is the current best upper estimate of 427 such that, for  $r \in V_k$ , value $(r)$  is the current best upper estimate of 427  $V(r)$ , parent $(r)$  is the parent node of r, and closed $(r) = 1$  if node  $V(r)$ , parent $r$ ) is the parent node of r, and closed $(r)=1$  if node 428<br>r has already been visited. Maps value, parent, and closed can be 429  $r$  has already been visited. Maps value, parent, and closed can be implemented as hash tables. 430

*Algorithm V.2 (A*<sup>∗</sup> *algorithm for* k*-BASP):* 431

1) [initialization] Set  $\mathcal{Q} = \{(s, h(s))\}$ , value $(s) = 0$ . 432<br>2) [expansion] Set  $(r, t) =$  DeleteMin( $\mathcal{Q}$ ) and set closed( $r$ ) = 1. 433

2) [expansion] Set  $(r, t)$  = DeleteMin( $\mathcal{Q}$ ) and set closed( $r$ ) = 1. 433<br> $\vec{r} \in \tilde{F}$ , then t is the optimal solution and the algorithm terminates. 434 If  $\vec{r} \in F$ , then t is the optimal solution and the algorithm terminates, 434 returning maps value, parent. Otherwise, for each  $\sigma \in \mathbb{V}$  for which 435 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 closed $(r') =$  $\Gamma(r, \sigma)$  is defined, set  $r' = \Gamma(r, \sigma)$ ,  $t' = t + \tilde{\eta}(r, \sigma)$ . If closed( $r'$ ) = 436<br>1, go to 3). Else, if value( $r'$ ) is undefined Insert( $\mathcal{Q}, (r, 't')$ ). Oth-<br>erwise if  $t' < \text{value}(r')$  set  $\text{value}(r') = t'$  parant( $r'$ ) = r and do erwise, if  $t' <$  value $(r')$ , set value $(r') = t'$ , parent $(r') = r$  and do 438 DecreaseKey $(\mathscr{Q},(r, 't')).$ )).  $439$ <br>o back to 2) otherwise no solution exists  $440$ 

3) [loop] If  $\mathcal{Q} \neq \emptyset$  go back to 2), otherwise no solution exists. 440<br>*Proposition V3*: Algorithm V2 terminates and returns the optimal 441

*Proposition V.3:* Algorithm V.2 terminates and returns the optimal 441 solution (if it exists), with a time-complexity not higher than Dijkstra's 442 algorithm on the extended graph  $\overline{\mathbb{G}}$ . 443<br>*Proof*: It is a consequence of the fact that heuristic h is admissible 444

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

Note that, at the end of Algorithm V.2,  $value(f)$  is the optimal value 446 the k-BASP and the optimal path from s to set F can be reconstructed 447 of the  $k$ -BASP and the optimal path from  $s$  to set  $F$  can be reconstructed from map parent. <sup>448</sup>

One possible limitation of Algorithm V.2 is that estimating  $K(\mathbb{B})$  449 m its definition can be difficult. A correct estimation of  $K(\mathbb{B})$  is 450 from its definition can be difficult. A correct estimation of  $K(\mathbb{B})$  is 450 critical for the efficiency of the algorithm. Indeed, if  $K(\mathbb{B})$  is overesti-451 critical for the efficiency of the algorithm. Indeed, if  $K(\mathbb{B})$  is overesti-<br>mated, the time complexity of the algorithm is higher than it would be mated, 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, 453<br>Algorithm V.2 is not correct since Proposition IV.4 does not hold. Here, 454 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 455 Algorithm V.2, such that  $k \le K(\mathbb{B})$ , but, in any case, allows to find the 456 optimal solution of the BASP First, we define the modified cost function 457 optimal solution of the BASP. First, we define the modified cost function 457  $W : \mathbb{V}_k \to \mathbb{R}$  as  $W(r) = V(r) + h(r)$ , where V is given by (6) and 458<br>*h* is the heuristic given by (8). If  $(\forall r \in \mathbb{V}_k)$   $\ell^+(r) < \ell^-(r)$ , then W is 459 h is the heuristic given by (8). If  $(\forall r \in \mathbb{V}_k) \ell^+(r) \leq \ell^-(r)$ , then W is 459<br>the solution of 460 the solution of

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

Indeed, following the same steps of the proof of Proposition IV.4, 461  $W(r) = V(r) + h(r) = \min_{r' \in \text{Prec}(r)} \{ V(r') + \eta(r', \vec{r}) + h(r) + 462 \}$ <br>  $h(r') = h(r') \} = \min_{r \in \mathbb{R}} \sum_{r'} \{ W(r') + \tilde{\eta}(r', \vec{r'}) \}$  Hence  $W(r) = 462$  $h(r') - h(r')$ } = min<sub>r'</sub> $\epsilon$ Prec(r){ $W(r') + \tilde{\eta}(r', \vec{r}')$ }. Hence,  $W(r)$  463 corresponds to the length of the shortest path from s to r on  $\tilde{\mathbb{G}}$ , 464 with arc length given according to  $\tilde{\eta}$ . If condition  $\ell^+(r) \leq \ell^-(r)$  is 465 with arc length given according to  $\tilde{\eta}$ . If condition  $\ell^+(r) \leq \ell^-(r)$  is 465 not satisfied for all  $r \in \mathbb{V}_k$ . (10) does not hold for all  $r \in \mathbb{V}_k$  and 466 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 467 proposition shows that we can still find a lower bound  $\hat{W}$  of W that 468 does correspond to the solution of an SP does correspond to the solution of an SP.

*Proposition V.4:* Let  $\hat{W}: \mathbb{V}_k \to \mathbb{R}$  be the solution of 470

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

where if  $\ell^+(r') \leq \ell^-(r')$  or  $|r'| < k$ ,  $\hat{\eta}(r', \vec{r}) = \tilde{\eta}(r, \vec{r})$ , otherwise 471<br>  $\hat{\eta}(r', \vec{r}) = b(r) - b(r')$ . Then  $(\forall r \in \mathbb{V})$ .  $\hat{\eta}(r', \vec{r}) = h(r) - h(r')$ . Then,  $(\forall r \in \mathbb{V}_k)$  472<br>
1)  $\hat{W}(r) < W(r)$ .

1)  $W(r) \leq W(r)$ ; 473<br>2)  $(\forall \bar{x} \in \mathbb{V}, \perp \hat{W}(\bar{x}) < \hat{W}(r)) \ell^+(\bar{x}) < \ell^-(\bar{x}) \rightarrow \hat{W}(r) = W(r)$  474

2)  $(\forall \bar{r} \in \mathbb{V}_k \mid \hat{W}(\bar{r}) \leq \hat{W}(r)) \ell^+(\bar{r}) \leq \ell^-(\bar{r}) \Rightarrow \hat{W}(r) = W(r).$  474<br>Proof: 1) For  $r \in \mathbb{V}$ . let  $n \in P$  be such that Suff,  $n \in \text{Proc}(r).$  475 *Proof:* 1) For  $r \in V_k$ , let  $p \in P_s$  be such that  $\text{Suff}_k p \in \text{Prec}(r)$ . 475<br>  $\ell^+(\text{Suff}_k p) \leq \ell^-(\text{Suff}_k p)$  in view of Proposition IV2 476 If  $\ell^+$ (Suff<sub>k</sub> p)  $\leq \ell^-$ (Suff<sub>k</sub> p), in view of Proposition IV.2, 476<br> $T(p\vec{r}) = T(p) + n$ (Suff<sub>k</sub> p,  $\vec{r}$ ), otherwise, obviously,  $T(p\vec{r}) \geq T(p)$ , 477  $T(p\vec{r}) = T(p) + \eta(\text{Suff}_k p, \vec{r})$ , otherwise, obviously,  $T(p\vec{r}) \geq T(p)$ . 477<br>Hence, in both cases, by the definition of  $\tilde{\eta}$  in (9),  $T(p\vec{r}) + h(r) > 478$ Hence, in both cases, by the definition of  $\tilde{\eta}$  in (9),  $T(p\tilde{r}) + h(r) \ge 478$ <br>  $T(p) + h(\text{Suff}_k, p) + \hat{\eta}(\text{Suff}_k, p, \tilde{r})$ . By contradiction. assume 479  $T(p) + h(\text{Suff}_k p) + \hat{\eta}(\text{Suff}_k p, \vec{r}).$  By contradiction,

445

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

498 Proposition V.4 implies that  $\hat{W}(r)$  is a lower bound of  $W(r)$  and that it corresponds to the length of the shortest path from s to r on 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 501 to (11), namely by the value of function  $\hat{\eta}$ . Hence,  $\hat{W}(f)$  can be 502 computed by Diikstra's algorithm (which is equivalent to compute V computed by Dijkstra's algorithm (which is equivalent to compute  $V$ 503 with A<sup> $*$ </sup> algorithm, with heuristic h). The algorithm that we are going to present is based on the following basic observation. If A<sup>∗</sup> algorithm 505 computes  $f^* = \argmin_{f \in \tilde{F}} W(f)$  by visiting only nodes for which 506  $\ell^+(r) < \ell^-(r)$ , then 2) of Proposition V.4 is satisfied for  $r = f^*$  and  $t^+(r) \leq t^-(r)$ , then 2) of Proposition V.4 is satisfied for  $r = f^*$  and  $\hat{W}(f^*) = W(f^*)$  is the optimal value of the k-BASP. If this is not the 507  $\dot{W}(f^*) = W(f^*)$  is the optimal value of the k-BASP. If this is not the 508 case we increase k by 1 and rerun the A<sup>\*</sup> algorithm. Note that the 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,  $K(\mathbb{B})$ <br>510 equals 1 only if no acceleration bounds are present and, in this case, the 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.

**algorithm V.5 (Adaptive A<sup>∗</sup> algorithm for k-BASP):** 

514 1) Set  $k = 2$ .<br>515 2) Execute A

2) Execute A<sup> $*$ </sup> 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  $k = k + 1$ <br>517 and repeat step 2). and repeat step 2).

518 Note that the algorithm does not compute the exact value  $K(\mathbb{B})$ .<br>519 Rather, it underestimates it. More precisely, it stops with the smallest 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  $k \le K(\mathbb{B})$  and 524 returns an optimal solution. 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  $k \leq K(\mathbb{B})$ 526  $\ell^+(r) \leq \ell^-(r)$  is satisfied for all r. Hence, there exists  $k \leq K(\mathbb{B})$ <br>527 for which the algorithm terminates. Let  $r \in \mathbb{V}_k$ , with  $\vec{r} \in F$  be the for which the algorithm terminates. Let  $r \in V_k$ , with  $\vec{r} \in F$  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 530 0), but, by definition,  $V(r)$  is the shortest time for reaching a node in 531  $F$ . 531  $F$ .  $\Box$ 

### 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

$n \mid k = 3 \mid k = 4 \mid k = 5 \mid k = 6 \mid k$			$k = 3   k = 4   k = 5   k = 6   \bar{k}$		
$100 80.4\% 18.0\% 1.6\% 0.0\% 86$			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 $					

48 (1)  $\frac{1}{2}$  (1)  $\frac{$ closer nodes have a higher connection probability. We multiplied the 539 obtained positions by factor  $10\sqrt{n}$ , in order to obtain the same average 540 node density independently of *n*. Then, we associated a random angle  $\theta_i$  541 node density independently of n. Then, we associated a random angle  $\theta_i$ to each node, obtained from a uniform distribution in  $[0, 2\pi]$ . In this way, 542 each node of the random graph is associated to a vehicle configuration. 543 each node of the random graph is associated to a vehicle configuration, consisting of a position and an angle. Set  $\tau(\theta_i) = [\cos \theta_i, \sin \theta_i]^T$ . 544 Each edge  $(i, j)$  is associated to a *Dubins path*, which is defined as the 545 shortest curve of bounded curvature that connects the configurations 546 shortest curve of bounded curvature that connects the configurations associated to nodes i and j, with initial tangent parallel to  $\tau(\theta_i)$  and 547 final tangent parallel to  $\tau(\theta_i)$ . We chose the minimum turning radius for 548 final tangent parallel to  $\tau(\theta_j)$ . We chose the minimum turning radius for 548<br>the path associated to edge (i, j) as  $r_{ii} = \min\{l((i, j))/(d(\theta_i, \theta_i))\}$  549 the path associated to edge  $(i, j)$  as  $r_{ij} = \min{\{\ell((i, j)) / (d(\theta_i, \theta_j))\}}$ , 2} 549<br>where  $d(x, y)$  is the angular distance between angles x and y. We set where  $d(x, y)$  is the angular distance between angles x and y. We set 550 the acceleration and deceleration bounds constant for all paths and 551 the acceleration and deceleration bounds constant for all paths and equal to 0.1 ms<sup>-2</sup>. The upper squared speed bound is constant for 552 each arc and given by  $2r$ , where r is the minimum curvature radius 553 of the path associated to the arc. In our tests, we used the adaptive 554 of the path associated to the arc. In our tests, we used the adaptive A<sup>∗</sup> algorithm (see Algorithm V.5). First, we ran simulations for ten 555 values of  $n$ , logarithmically spaced between 100 and 1000. For each 556  $n$ , we generated 50 different graphs, and for each one of them, we  $557$ ran ten simulations, randomly choosing source and target nodes. Fig. 5 558 shows the mean values and the distributions of the computational times 559 of Algorithm V.5 and it also shows the mean computational times of 560 Algorithm V.2 with  $k$  computed as in (5). Note that the mean times of 561 Algorithm V.2 are at least one order of magnitude higher than those of 562 Algorithm V.5. Table I shows, for each  $n$ , the percentages of  $k$  values 563 returned by Algorithm V.5, and the mean value k of k computed as 564 in (5). Note that the values obtained with (5) are on average 54.8 times 565 in  $(5)$ . Note that the values obtained with  $(5)$  are on average 54.8 times larger than those returned by Algorithm V.5. 566

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

	travel time gain [%] with respect to BASP without acceleration bounds			
	30 travel time gain $[\%]$ with respect to SP	40	50	60

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.

Fig. 2. These formula methods in the set of t 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 581 paths *p*<sup>∗</sup> and *ρ*̂, that is, for each value of the acceleration and deceleration bounds. we report the relative percentage difference  $100 \frac{T_{\text{B}}(p)-T_{\text{B}}(p^*)}{T_{\text{B}}(p^*)}$ bounds, we report the relative percentage difference  $100 \frac{T_B(\hat{p}^*) - T_B(p^*)}{T_B(\hat{p}^*)}$ 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 585 and deceleration bounds increase, the travel time difference between the 586 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<br>588 (bottom), we compare the length of paths  $p^*$  and  $\tilde{p}$  showing how the 588 (bottom), we compare the length of paths  $p^*$  and  $\tilde{p}$  showing how the 588 and pSP solution tends to differ from the SP with edge costs equal to their 589 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. 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  $\alpha^+ = 0.28$  ms<sup>-2</sup> and 598  $\alpha^- = -0.18$  ms<sup>-2</sup>. The speed bounds are constant for each arc but  $\alpha^- = -0.18$  ms<sup>-2</sup>. The speed bounds are constant for each arc but vary among different arcs, according to the associated path curvatures. vary among different arcs, according to the associated path curvatures, 600 and they take values on interval  $[0.1, 1.7]$  ms<sup>-1</sup>. The arc lengths take 601 values in  $[0.2, 18]$  m and have an average value of 4.2 m. We ran 1000 601 values in  $[0.2, 18]$  m and have an average value of 4.2 m. We ran 1000 602 simulations by randomly choosing source and the target nodes. The 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 607 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 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 610 the relative percentage differences over 1000 tests. In the first case, we 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% mean gain and the 25% best performing paths  $p^*$  had a 14.16% mean 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 621

*Proposition A.1:* Let  $\mu, \alpha : [0, +\infty) \to \mathbb{R}^+$ , for  $i \in \{1, 2\}$ , let  $F_i$  622 the solution of the differential equation (2) where  $F_i$  replaces  $F$  623 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  $\lambda$  be such that 624  $\mu(\bar{\lambda}) = \int_0^{\bar{\lambda}} \alpha(\lambda) d\lambda$ . Then,  $(\forall \lambda \geq \bar{\lambda})$ <br>Proof: Without loss of generali

 $\bar{\lambda}$ ) =  $\int_0^{\infty} \alpha(\lambda) d\lambda$ . Then,  $(\forall \lambda \ge \bar{\lambda}) F_1(\lambda) = F_2(\lambda)$ . 625<br>*Proof:* Without loss of generality, assume that  $w_{0,1} \ge w_{0,2}$ . This 626 implies that  $(\forall \lambda \geq 0) F_1(\lambda) \geq F_2(\lambda)$ . Indeed, assume by contradic-<br>tion that there exists  $\overline{\lambda}$  such that  $F_1(\overline{\lambda}) < F_2(\overline{\lambda})$ , then, by conti- 628 tion that there exists  $\lambda$  such that  $F_1(\lambda) < F_2(\lambda)$ , then, by conti- 628<br>puity of E, and E, this implies that there exists  $\lambda \leq \overline{\lambda}$  such that nuity of  $F_1$  and  $F_2$ , this implies that there exists  $\lambda \leq \lambda$  such that 629  $F_1(\lambda) = F_2(\lambda)$ , thus  $(\forall \lambda \ge \lambda) F_1(\lambda) = F_2(\lambda)$ , since, for  $\lambda \ge \lambda$ <br> $F_1(\lambda)$  and  $F_2(\lambda)$  solve the same differential equation with the same 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 initial condition at  $\lambda = \lambda$ , contradicting the assumption. Furthermore, 632<br>note that  $(\exists \tilde{\lambda} \in (0, \tilde{\lambda}) \setminus F_2(\tilde{\lambda}) = u(\tilde{\lambda})$ . Indeed, if by contradiction 633 note that  $(\exists \lambda \in (0, \lambda]) F_2(\lambda) = \mu(\lambda)$ . Indeed, if by contradiction 633<br>( $\forall \lambda \in (0, \overline{\lambda})$ ) E ( $\lambda$ )  $\leq \mu(\lambda)$  then  $(\forall \lambda \in (0, \overline{\lambda})$ ) E'( $\lambda$ ) =  $\alpha(\lambda)$  so that  $(\forall \lambda \in (0, \bar{\lambda})$   $F_2(\lambda) < \mu(\lambda)$ , 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-<br>time. Hence  $(\bar{\lambda}) - \bar{F}(\hat{\lambda}) = F(\hat{\lambda})$ , which contradicts the assumption. Hence,  $(\exists \lambda \in \mathbb{R}^+) F_2(\lambda) = F_1(\lambda) = \mu(\lambda)$ , and consequently, 636<br>( $\forall \lambda > \hat{\lambda}$ )  $F_1(\lambda) = F_2(\lambda)$  which implies the thesis being  $\bar{\lambda} > \hat{\lambda}$  $(\forall \lambda \ge \lambda) F_1(\lambda) = F_2(\lambda)$ , which implies the thesis, being  $\lambda \ge \lambda$ .  $\square$ <br>For  $n \in P$ ,  $\lambda \in [0, \ell(n)]$ , we set  $\mathcal{W}(1) = uv$ , where  $uv$  is the solution 637

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

*Proposition A.2 1):* Let  $p_1, p_2, q \in P$ , be such that  $p_1q, p_2q \in P$ , 642 then  $(\forall \lambda \geq \ell^+(q))$   $\mathscr{W}_{p_1q}(\ell(p_1) + \lambda) = \mathscr{W}_{p_2q}(\ell(p_2) + \lambda).$  643<br>2) Let  $p, q_2, q_1 \in P$ , be such that  $pq_1, pq_2 \in P$ , then  $(\forall \lambda \leq \lambda)$ 

2) Let  $p, q_2, q_1 \in P$ , be such that  $pq_1, pq_2 \in P$ , then  $(\forall \lambda \leq \theta_1 \in \ell^-(p))$   $\mathscr{W}_{pq_1}(\lambda) = \mathscr{W}_{pq_2}(\lambda)$ .  $\ell^-(p)$ )  $\mathscr{W}_{pq_1}(\lambda) = \mathscr{W}_{pq_2}(\lambda)$ . 645<br>*Proof:* We only prove 1), the proof of 2) is analogous. Note 646

that, for  $\lambda \geq 0$ ,  $\mathscr{W}_{p_1q}(\lambda + \ell(p_1)) = \min\{F_1(\lambda), B(\lambda)\}$ ,  $\mathscr{W}_{p_2q}(\lambda + 647)$ <br> $\ell(p_2) = \min\{F_2(\lambda), B(\lambda)\}$ , where  $F_1$  and  $F_2$  are the solution of (2) 648  $\ell(p_2)$  = min{F<sub>2</sub>( $\lambda$ ),  $B(\lambda)$ }, where F<sub>1</sub> and F<sub>2</sub> are the solution of (2) 648<br>with  $\mu = \mu^+$  and initial conditions  $w_0 = \mathcal{W}_+$  ( $\ell(n_1)$ ) and  $w_0 =$  649 with  $\mu = \mu^+$  and initial conditions  $w_{0,1} = \mathcal{W}_{p_1}(\ell(p_1))$  and  $w_{0,2} = 649$ <br> $\mathcal{W}_{\alpha}(\ell(p_2))$ , respectively, and B is the solution of (3) with  $\mu = \mu^+$ , 650  $\mathscr{W}_{p_2}(\ell(p_2))$ , respectively, and *B* is the solution of (3) with  $\mu = \mu^+$ . 650 By Proposition A.1, for  $\lambda > \ell^+(q)$ ,  $F_1(\lambda) = F_2(\lambda)$ . Consequently, 651 By Proposition A.1, for  $\lambda \geq \ell^+(q)$ ,  $F_1(\lambda) = F_2(\lambda)$ . Consequently,  $(\forall \lambda > \ell^+(q))$   $\mathcal{W}_{-}(l(p_1) + \lambda) = \mathcal{W}_{-}(l(p_2) + \lambda)$ .  $(\forall \lambda \geq \ell^+(q))$   $\mathscr{W}_{p_1q}(\ell(p_1) + \lambda) = \mathscr{W}_{p_2q}(\ell(p_2) + \lambda).$ 652

#### *A. Proof of Proposition IV.2* 653

Let  $\Psi$  be defined as in (1a), then  $T(p_1 t\sigma) - T(p_1 t) = \int_0^{\ell(p_1 t\sigma)} \Psi$  654  $(\mathscr{W}_{p_1 t\sigma}(\lambda))d\lambda - \int_0^{\ell(p_1t)} \Psi(\mathscr{W}_{p_1t}(\lambda))d\lambda = \int_{\ell(p_1)+\ell^-(t)}^{\ell(p_1t\sigma)} \Psi(\mathscr{W}_{p_1t\sigma}(\lambda))$  655  $d\lambda - \int_{\ell(p_1)}^{\ell(p_1t)} \Psi(\mathscr{W}_{p_1t}(\lambda))d\lambda$ , where we used that, by 2) of 656<br>Proposition  $\lambda$ ,  $\chi(\lambda) \leq \ell(p_1) + \ell(-\ell)$ Proposition A.2,  $(\forall \lambda \leq \ell(p_1) + \ell^-(t)) \Psi(\mathscr{W}_{p_1 t \sigma}(\lambda)) = \Psi(\mathscr{W}_{p_1 t}(\lambda)).$  657 Similarly,  $T(p_2 t\sigma)$  –  $T(p_2 t)$  =  $\int_{\ell(p_2)+\ell^-(t)}^{\ell(p_2 t\sigma)} \Psi(\mathscr{W}_{p_2 t\sigma}(\lambda))d\lambda$  – 658  $\int_{\ell(p_2)+\ell^{-}(t)}^{\ell(p_2t)} \Psi(\mathscr{W}_{p_2t}(\lambda))d\lambda$ . Moreover, by 1) of Proposition A.2, we 659 have that  $(\forall \lambda \geq \ell^+(t\sigma))$   $\mathscr{W}_{p_1t\sigma}(\ell(p_1) + \lambda) d\lambda = \mathscr{W}_{p_2t\sigma}(\ell(p_2) + \lambda) d\lambda$  660<br>and  $(\forall \lambda \geq \ell^+(t))$   $\mathscr{W}_{p_1t}(\ell(p_1) + \lambda) d\lambda = \mathscr{W}_{p_2t}(\ell(p_2) + \lambda) d\lambda$ , 661  $\mathscr{W}_{p_1t}(\ell(p_1) + \lambda)d\lambda = \mathscr{W}_{p_2t}(\ell(p_2) + \lambda)d\lambda$ , 661<br>to  $-\Gamma(n_1t) = \Gamma(n_2t\sigma) - \Gamma(n_2t)$ , since 662 which imply that  $T(p_1t\sigma) - T(p_1t) = T(p_2t\sigma) - T(p_2t)$ , since 662<br> $\ell^+(t) \leq \ell^-(t)$ , and as noticed in Section IV,  $\ell^+(t\sigma) \leq \ell^+(t)$ .  $\ell^+(t) \leq \ell^-(t)$ , and as noticed in Section IV,  $\ell^+(t\sigma) \leq \ell^+(t)$ .

#### 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 andM. A. Oskoei, "Mobile robot path planning by RRT<sup>∗</sup> in dynamic 668 environments," *Int. J. Intell. Syst. Appl.*, vol. 7, no. 5, pp. 24–30, 2015. 669
- 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. 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. **Q3** 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.
	- [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.
	- [8] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A<sup>\*</sup>," *Artif.* 685 *Intell.*, vol. 155, no. 1-2, pp. 93–146, 2004.
	- 686 [9] D. Ferguson and A. Stentz, "Field D<sup>∗</sup>: An interpolation-based path planner and replanner," *Springer Tracts Adv. Robot.*, vol. 28, pp. 239–253, 2007. 687 and replanner," *Springer Tracts Adv. Robot.*, vol. 28, pp. 239–253, 2007.
	- 19 (1) Excels the same of the 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. 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<br>Lett., vol. 83, pp. 61–66, 2015. *Lett.*, vol. 83, pp. 61–66, 2015.
- [14] T. Faulwasser, V. Hagenmeyer, 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**<br>D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. 704
- [15] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. 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 algorithms," 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.
- [19] S. Edelkamp and S. Schroedl, *Heuristic Search: Theory and Applications*. 716 Elsevier Science, 2011.