Computational Optimization and Applications
https://doi.org/10.1007/s10589-022-00403-w

Shortest path with acceleration constraints: complexity and approximation algorithms

S. Ardizzoni[1] · L. Consolini[1] · M. Laurini[1] · M. Locatelli[1]

Received: 4 February 2022 / Accepted: 22 July 2022

Abstract

We introduce a variant of the Shortest Path Problem (SPP), in which we impose additional constraints on the acceleration over the arcs, and call it Bounded Acceleration SPP (BASP). This variant is inspired by an industrial application: a vehicle needs to travel from its current position to a target one in minimum-time, following pre-defined geometric paths connecting positions within a facility, while satisfying some speed and acceleration constraints depending on the vehicle position along the currently traveled path. We characterize the complexity of BASP, proving its NP-hardness. We also show that, under additional hypotheses on problem data, the problem admits a pseudo-polynomial time-complexity algorithm. Moreover, we present an approximation algorithm with polynomial time-complexity with respect to the data of the original problem and the inverse of the approximation factor ϵ. Finally, we present some computational experiments to evaluate the performance of the proposed approximation algorithm.

Keywords Shortest path · Speed planning · Complexity · Approximation algorithms

1 Introduction

The Shortest Path Problem (SPP in what follows) is one of the best known within the field of combinatorial optimization. Let $G=(V,A)$ be a directed graph and c_{ij} be the cost of an arc $(i,j) \in A$. Let $n=|V|$ and $m=|A|$. Let $o,d \in V$, $o \neq d$, be an origin and a destination node, respectively, and let $\mathcal{P}_{od}$ be the set of all directed paths in G from o to d. Each $P \in \mathcal{P}_{od}$ is a subset of A made up of adjacent arcs, the first one starting at o and the last one ending at d. In the SPP the minimum cost path between o and d is searched for, that is, formally:

✉ M. Locatelli
locatell@ce.unipr.it

[1] Dipartimento di Ingegneria e Architettura, Università degli Studi di Parma, Parco Area delle Scienze, 181/A, Parma, Italy

Published online: 10 August 2022

$$\min_{P\in\mathcal{P}_{od}} \sum_{(i,j)\in P} c_{ij} .$$

It is well known that SPP is solvable in polynomial time for nonnegative costs, in particular with $O(m + n\log(n))$ operations by the Fibonacci heap implementation of Dijkstra's algorithm, and also if no negative cycle exists, for instance by the Bellman-Ford algorithm. In both these cases, one can restrict the search of optimal solutions to elementary paths (paths with no node repetitions). If negative cycles are present, then the problem has been proved to be NP-hard.

After the introduction of SPP, many variants have been proposed, together with the related complexity results and solution approaches. In what follows, we discuss some of these variants, warning the reader that the literature about such variants is so vast that the list is certainly incomplete.

Some variants do not add further input data with respect to those of SPP, but change the objective function. Many of these variants are reviewed in [21]. These include *bottleneck* SPP, where the objective function is the largest cost of an arc along the path, that is, $\max_{(i,j)\in P} c_{ij}$, *balanced* SPP, where the objective function is the difference between the largest and smallest costs of arcs along the path, that is, $\max_{(i,j)\in P} c_{ij} - \min_{(i,j)\in P} c_{ij}$, *minimum deviation* SPP, where the objective function is the sum of the difference between the largest cost among all arcs of the path and the cost of each arc along the path, that is, $\sum_{(h,k)\in P}[\max_{(i,j)\in P} c_{ij} - c_{hk}]$, *k-sum* SPP, where the objective function is the sum of the k largest costs along the path. While some of these problems are solvable in polynomial time, such as bottleneck SP (see [11, 20]), some others are NP-hard, such as some variants of k-sum SPP (see [13]). In [7], the SPP with forbidden paths is addressed, where the shortest path is searched for with the additional constraint that some sub-paths cannot be part of feasible solutions. In k-SPP not only the shortest path is searched for, but also all paths from the second shortest up to the k-th shortest one (see, for instance, [2] and references therein).

Other variants add additional input values to the description of the problem. For instance, in resource constrained SPPs, given resources $k \in \{1,\dots,K\}$, each with a limited availability γ_k and a consumption value r_{ij}^k along each arc, we search for a minimum cost path P which satisfies all the resource constraints

$$(\forall k \in \{1,\dots,K\}) \sum_{(i,j)\in P} r_{ij}^k \leq \gamma_k .$$

Resource constrained SPPs have been proved to be NP-hard (see, for instance, [14, 15]). A detailed discussion of these problems and the related solution approaches can be found in [12].

Another interesting variant of SPP, is *time-dependent* SPP. Here, the cost associated to an arc is the time needed to traverse the arc, and such time depends on the departure time along the arc. Such variant is particularly important for road networks, where the time needed to traverse an arc varies according to traffic conditions. In this case, given a time horizon T, the cost associated to an arc is not a fixed value but is a function $c_{ij} : [0,T] \to \mathbb{R}^+$, where $c_{ij}(t)$ is the time needed to traverse arc (i, j) when departing from i at time t. In these problems the first-in first-out (FIFO) property is usually assumed. This states that along any arc (i, j) an earlier arrival at node j can never be attained by a later departure at node i (that is, for $t' > t$ it always holds that $t' + c_{ij}(t') > t + c_{ij}(t)$). Many variants of this problem are presented, for instance, in [6]. In [18], it is shown that a variant where there is a constraint on the waiting time at each node is NP-hard. Further variants are discussed in [16]. In these variants a penalty or a limit is imposed on the total waiting time spent at a given subset of the nodes. It is proved that some variants are polynomially solvable, while some others are NP-hard depending on the subset of nodes, on the fact that a penalization or a limit is imposed, and on the magnitude of the penalty parameter or of the waiting limit parameter.

Another interesting variant is SPP with time windows. In this case we associate to each node $i \in V$ an interval $[a_i, b_i]$, and to each arc (i, j) a time t_{ij} to traverse it. Only paths in $\mathcal{P}_{od}$ where each node of the path is visited within the allowed time window are feasible. The case where only elementary paths are feasible has been proven to be strongly NP-hard in [10]. The problem is still NP-hard if we remove such restriction and, thus, if we allow multiple visits to the same node. However, in such case pseudo-polynomial time algorithms have been proposed (see, for instance, [8]). A variant with additional costs associated to the departure times at the different nodes of the path is studied in [17]. In fact, a polynomial time algorithm exists if the FIFO property holds (see [9]). Moreover, if waiting is allowed at a node, then each instance for which the FIFO property does not hold can be transformed into an equivalent one for which the property holds, thus making the problem solvable in polynomial time (see [19]).

In previous work [1], we introduced a further variant of SPP called BASP (Bounded Acceleration SP) and proposed a solution algorithm. The interest for BASP arises from an industrial application, namely the optimization of automated guided vehicles (AGVs) motions in automated warehouses (see also the description in Sect. 2). Typically, AGVs follow pre-assigned routes associated to a graph, in which nodes represent operating positions and arcs represent connecting paths. AGVs motions must satisfy constraints on maximum speed, and tangential and transversal accelerations. BASP consists in finding the path in $\mathcal{P}_{od}$ and the speed profile that allows travelling in minimum-time. The speed profile must satisfy maximum speed, acceleration and deceleration constraints, associated to each arc. In [1], we presented a solution algorithm (adaptive A*) for k-BASP, a subclass of BASP. Roughly speaking, a BASP instance is a k-BASP, with $k \in \mathbb{N}$, if the maximum number of nodes of a path that can be traveled with a speed profile of maximum acceleration, followed by one of maximum deceleration, starting and ending with null speed, without violating the maximum speed constraint, is smaller than k (see [1] for details). In [1], we proved that k-BASP has polynomial time-complexity with respect to the graph size. The algorithm introduced in [1] computes the optimal trajectory between a pair of nodes and adaptively determines the value of k.

In some applications, AGVs can freely move within the facility, without having to follow any predetermined circuit. In such cases, the motion planning problem is usually addressed employing environmental representations such as cell decomposition methods. Among these, [5] presents an algorithm based on a modification of

Dijkstra's algorithm, in which arcs weights depend on previously visited edges. Note that k-BASP in [1] has some similarities with the problem discussed in [5]. Indeed, in both problems, the incremental for adding an edge to a path does not depend on the complete path, but only on the k last visited nodes. However, the problem addressed by Cowlagi and Tsiotras [5] is different from BASP. Indeed, the goal of the problem in [5] is to obtain a feasible path taking into account the vehicle maximum curvature radius. On the other hand, in our work, we focus on selecting the optimal path among a set of possible paths which are already known to be feasible, while, at the same time, obtaining the optimal speed profile. Moreover, we do not assume that the incremental cost depends only on the last k visited nodes, but present conditions on problem data under which this property holds.

In this paper we discuss BASP in further detail, presenting two novel complexity results. Namely, we show that BASP is NP-hard and that, under additional assumptions, BASP admits a pseudo-polynomial time algorithm. We also introduce an ϵ -approximation algorithm based on the discretization of the admissible speeds at the nodes of the graph and show computational experiments comparing the ϵ-approximation algorithm with the solution algorithm presented in [1].

The paper is structured as follows. In Sect. 2 we describe and motivate BASP. In Sect. 3 we discuss how to compute optimal speed profiles along a single arc (with given initial and final speeds), and along a fixed path belonging to $\mathcal{P}_{od}$. In Sect. 4 we present the aforementioned complexity results. More precisely, in Sect. 4.1 we prove that BASP is NP-hard, while in Sect. 4.2 we prove that, under the assumption of integer data, BASP admits a pseudo-polynomial time algorithm. In Sect. 5 we present the ϵ-approximation algorithm. Finally, in Sect. 6 we present some computational experiments.

2 Problem description and motivation

In this section, we describe a new variant of SPP. The following values are associated to each arc $(i,j) \in A$:

- the length ℓ_{ij};
- the maximum speed $v_{ij}^{\max}$ which can be reached along the arc;
- the minimum acceleration (or maximum deceleration) $a_{ij}^{\min} < 0$ and maximum acceleration $a_{ij}^{\max} > 0$ along the arc.

Moreover, a zero initial speed is assigned to node o and a zero final speed is assigned to node d. We would like to select a path $P \in \mathcal{P}_{od}$ minimizing the time needed to run along the path by fulfilling the maximum speed, the maximum and minimum acceleration constraints along the arcs, and the boundary zero conditions on o and d. Note that SPP is a special case of this problem. Indeed, SPP turns out to be equivalent to the case $a_{ij}^{\max} = +\infty$ and $a_{ij}^{\min} = -\infty$ for all arcs $(i,j) \in A$. In this case, the speed can be changed instantaneously, so that the running time along an arc is minimized by traversing it at the maximum allowed speed along the arc. Therefore, the minimum time to traverse arc (i,j) is $c_{ij} = \frac{\ell_{ij}}{v_{ij}^{\max}}$ and the problem becomes a standard SPP with such costs c_{ij} associated to the arcs. Since SPP corresponds to the case of *unbounded* acceleration limits, the proposed variant of SPP is also called Bounded Acceleration SP (BASP in what follows). In the search for an optimal solution of BASP, we should not only search for an optimal path in $\mathcal{P}_{od}$, but we should also define the speed profile along such path. As we will see later on in Sect. 3, the minimum-time speed profile along an arc (i,j) is fully determined by the initial speed v_i at node i and by the final speed v_j at node j. Thus, the speed v_i at each node i traversed by a path $\mathcal{P}_{od}$ is also part of the decision process. Such speeds must fulfill a continuity constraint, that is, if arc (i,j) is followed by arc (j,k) along the path, the final speed along arc (i,j) must be equal to the initial speed along arc (j,k). Note that the speeds at the origin node o and at the destination node d are fixed in advance.

Before proceeding, we further motivate the interest for the BASP variant of SPP. As previously mentioned, the interest comes from an industrial application. In automated warehouses, an AGV is required to pick some good up at some point of the warehouse and deliver it at some other point. The AGV moves along predefined paths and is allowed to choose among different routes at some exchange points. Formally, the exchange points as well as the points where goods are picked up and delivered represent the nodes of the graph, while the predefined routes correspond to the arcs of the graph (see, for instance, Figs. 8, 13). Speed and acceleration limits differ across the different routes. For instance, if a route is a straight line, its maximum speed is higher than the maximum speed allowed along a curved route. Moreover, different speed limits may also be imposed at different points of the warehouse. For instance, if a route lies in a part of the warehouse where also human operators are working, then, for safety reasons, a lower speed limit along this route is imposed with respect to another route lying in a part of the warehouse where human operators are not allowed to work. Due to various reasons, also acceleration bounds may differ from arc to arc. For instance, in the same warehouse, flooring materials can vary from location to location. To avoid wheel slipping, we need to set maximum acceleration bounds depending on the floor frictional force, that varies according to the flooring material. Moreover, the presence of ramps along some arcs implies different acceleration and deceleration bounds. Finally, to reduce lateral oscillations, we should impose lower acceleration bounds along high-curvature connecting paths.

Remark 2.1 In the problem description we imposed a constant speed limit $v_{ij}^{\max}$ along each arc (i,j). In a more realistic setting the speed limit should be a function of the position along a given route:

$$v_{ij}^{\max} : [0, \ell_{ij}] \to \mathbb{R}^+.$$

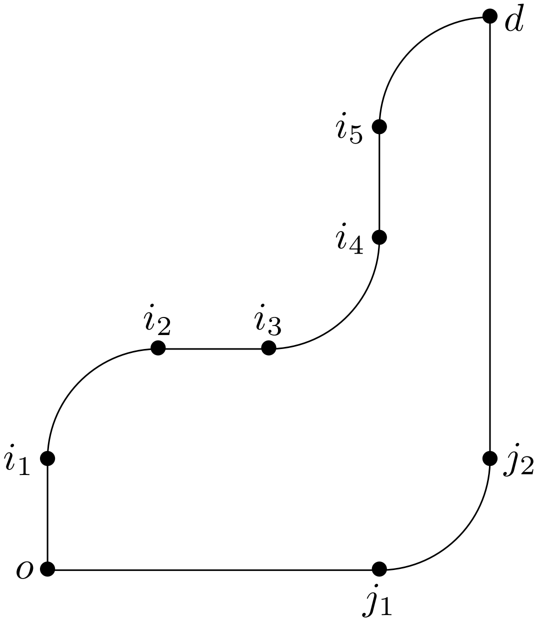
For each $s \in [0, \ell_{ij}]$, $v_{ij}^{\max}(s)$ is the maximum allowed speed at position s along the route represented by arc (i,j). In particular, along a curve the speed limit varies with the curvature ray at each position along the curve. For ease of exposition, we only discuss the case of a constant speed limit along an arc (although different from arc to arc). However, at the cost of some additional technicalities, the following discussion can be extended also to the more realistic case of variable speed limits along each arc.

Remark 2.2 In a real industrial scenario, the AGV may encounter moving obstacles, such as human operators and/or other AGVs. For safety, an AGV typically halts or slows down if it perceives the presence of a human operator. When the obstacle is no longer sensed, the AGV can compute a new motion by solving a new instance of BASP, starting from its current location.

The presence of various AGVs leads to a variant of the Multi-Agent Path Finding (MAPF) problem (see, for instance, [22]), with speed and acceleration constraints. Due to the simultaneous planning of multiple AGVs, this problem is quite different from BASP (and in general much more complex), and is outside the scope of the present work. However, note that some solution approaches for standard MAPF (that does not consider velocity or acceleration bounds), such as conflict-based search, make use of sub-procedures that involve the solution of a number of standard SPP problems. Similarly, one could guess that BASP solutions could be used as basic building block to solve MAPF with speed and acceleration bounds.

In order to better appreciate the difference between SPP and BASP we can also consider the example illustrated in Fig. 1. While path $o \to i_1 \to i_2 \to i_3 \to i_4 \to i_5 \to d$ is shorter than path $o \to j_1 \to j_2 \to d$, the latter is faster in view of the lower number of curves along the path.

Fig. 1 Two distinct paths from node o to node d: the shorter path is not the faster one



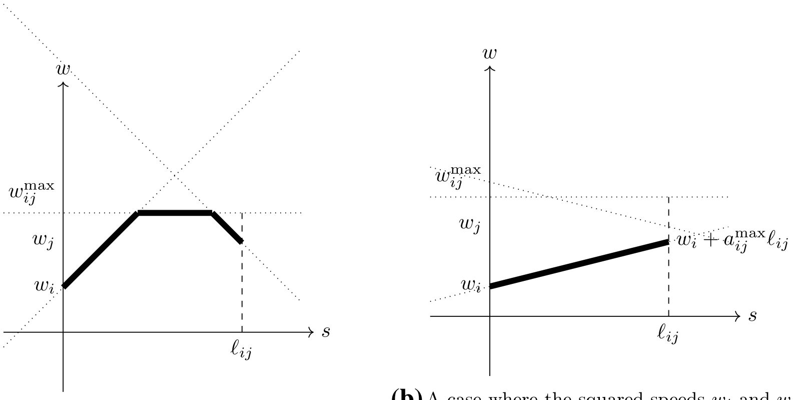
3 Minimum traveling time along arcs and paths

As previously pointed out, for a given arc (i, j), we are able to compute the maximum speed at which we can traverse the arc and, consequently, the minimum time needed to traverse it, as soon as we know the initial and final speeds v_i and v_j. Indeed, the maximum squared speed at each position $s \in [0, \ell_{ij}]$ along the arc is given by the following piecewise-linear function:

$$w_{ij}(s; w_i, w_j) = \min\left\{ w_{ij}^{\max}, w_i + 2a_{ij}^{\max} s, w_j + 2a_{ij}^{\min}(s - \ell_{ij}) \right\}, \tag{1}$$

where here and in what follows, w denotes the squared speed (so $w_i = v_i^2$, $w_j = v_j^2$, and so on). The result is illustrated in Fig. 2a. Starting at node i with squared speed w_i, the speed is increased with the maximum possible acceleration $a_{ij}^{\max}$, until the maximum allowed squared speed $w_{ij}^{\max}$ along the arc is reached. Such maximum speed is maintained as long as possible (null acceleration) and, finally, the speed is decreased with the maximum deceleration $a_{ij}^{\min}$ in order to reach squared speed w_j at node j. Note that it might be possible that the maximum speed along the arc is not reachable. In such case, first the speed is increased with maximum acceleration $a_{ij}^{\max}$ and then decreased with maximum deceleration $a_{ij}^{\min}$ to reach the final squared speed w_j (no constant speed portion is present in the maximum speed profile). Consequently, the minimum time to traverse arc (i, j) is the following function of the two (squared) speeds w_i and w_j

$$c_{ij}(w_i, w_j) = \int_{s=0}^{\ell_{ij}} \frac{ds}{\sqrt{w_{ij}(s; w_i, w_j)}}, \tag{2}$$



(a) Maximum (squared) speed profile along arc (i, j) with initial and final squared speed equal to w_i and w_j, respectively.

(b) A case where the squared speeds w_i and w_j are not feasible.

Fig. 2 Squared speed profiles along an arc

whose solution can be derived in closed form. It is worthwhile to remark that w_i and w_j cannot be arbitrary values. Indeed, besides the obvious constraints $w_i, w_j \leq w_{ij}^{\max}$, it must also hold that

$$w_i + 2a_{ij}^{\max}\ell_{ij} \geq w_j, \quad w_j - 2a_{ij}^{\min}\ell_{ij} \geq w_i. \tag{3}$$

If, for instance, the first inequality is not fulfilled, then it is not possible to reach the squared speed w_j at node j by starting with squared speed w_i at node i even by accelerating as much as possible (acceleration $a_{ij}^{\max}$). This is illustrated in Fig. 2b. In this case we set $c_{ij}(w_i, w_j) = +\infty$.

While we reported above the formula for the minimum time needed to traverse an arc (i, j), given the initial and final squared speeds w_i and w_j, we further point out that a slightly more complicated formula can be derived to compute the minimum time to traverse a full path in $\mathcal{P}_{od}$. In particular, let $i_0 = o \to i_1 \to \ldots \to i_{p-1} \to i_p = d$ be a path of length p in $\mathcal{P}_{od}$. Let $s_0 = 0$ and $s_h = \sum_{r=1}^{h} \ell_{i_{r-1}i_r}$ be the overall length of the first h arcs of the path, $h \in \{1, \ldots, p\}$. Next, let us define recursively a function F as follows

$$(\forall s \in [s_0, s_1])\ F(s) = \min\left\{ w_{i_0 i_1}^{\max}, 2a_{i_0 i_1}^{\max} s \right\},$$
$$(\forall h \in \{1, \ldots, p-1\})\ (\forall s \in [s_h, s_{h+1}])\ F(s) = \min\left\{ w_{i_h i_{h+1}}^{\max}, F(s_h) + 2a_{i_h i_{h+1}}^{\max}(s - s_h) \right\},$$

and a further function B as follows

$$(\forall s \in [s_{p-1}, s_p])\ B(s) = \min\left\{ w_{i_{p-1} i_p}^{\max}, 2a_{i_{p-1} i_p}^{\min}(s - s_p) \right\},$$
$$(\forall h \in \{1, \ldots, p-1\})\ (\forall s \in [s_{h-1}, s_h])\ B(s) = \min\left\{ w_{i_{h-1} i_h}^{\max}, B(s_h) + 2a_{i_{h-1} i_h}^{\min}(s - s_h) \right\}.$$

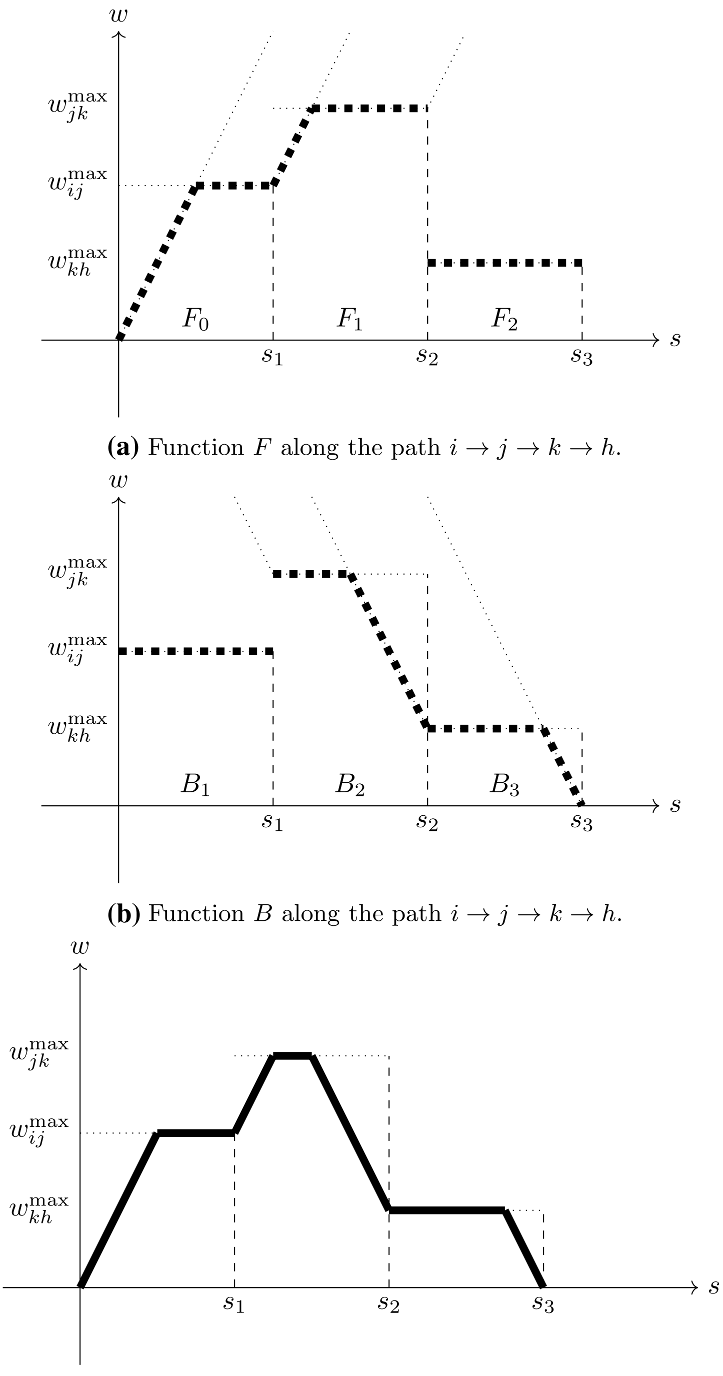
Then, it can be proved (see, for instance, [4] for a proof under more general assumptions) that the optimal (squared) speed profile is

$$(\forall s \in [s_0, s_p])\ W(s) = \min\{F(s), B(s)\},$$

so that the minimum time to travel along the given path is

$$\int_{s=0}^{s_p} \frac{ds}{\sqrt{W(s)}}.$$

The result is illustrated in Fig. 3a–c. In Fig. 3a we notice that F_0 (function F over the interval $[s_0, s_1]$) is obtained by starting at node o with zero speed and increasing the speed with maximum acceleration $a_{ij}^{\max}$ until the maximum squared speed $w_{ij}^{\max}$ is reached, and then keeping this speed until the end of the arc (as it is the case in the figure) or, alternatively, until the end of the arc is reached (in which case the maximum squared speed $w_{ij}^{\max}$ is not reached). Next, F_1 (function F over the interval $[s_1, s_2]$) is obtained similarly but with initial speed $F(s_1)$, while F_2 (function F over the interval $[s_2, s_3]$) is constant and equal to the maximum squared speed $w_{kh}^{\max}$ since $F(s_2) > w_{kh}^{\max}$. In Fig. 3b we notice that B_3 (function B over the interval $[s_2, s_3]$) is



(a) Function F along the path $i \to j \to k \to h$.

(b) Function B along the path $i \to j \to k \to h$.

(c) Maximum (squared) speed profile along the path $i \to j \to k \to h$.

Fig. 3 Construction of a maximum (squared) speed profile along a path

obtained in a way completely similar to F_0. Moving backward, we start from the final node $d \equiv h$ with squared speed $w_h = 0$, and we increase the speed with the maximum deceleration value $a^{\min}_{kh}$ until either we reach the maximum allowed squared speed $w^{\max}_{kh}$ along the arc, in which case we keep such speed until the beginning of the arc at node k, or we reach the beginning of the arc without reaching the maximum speed along the arc (in the figure we are in the first situation). Function B_2 (function B over the interval $[s_1, s_2]$) is obtained similarly but with initial speed at node k equal to $B(s_2)$. Finally, B_1 (function B over the interval $[s_0, s_1]$) is constant and equal to the maximum squared speed $w^{\max}_{ij}$ since $B(s_1) > w^{\max}_{ij}$. The optimal (squared) speed profile is illustrated in Fig. 3c and is obtained as the point-wise minimum of the functions F and B. While Figs. 2a, b and 3a, c give an intuitive illustration of the optimal speed profiles both for the case of a single arc (i, j) and for the case of a full path from o to d, we point out that these results can be derived as special cases of a more general result presented in [4] for problems with upper speed limits depending on the position along the arcs, as discussed in Remark 2.1.

We also make the following remark which states two properties of the optimal speed profile and will be useful later on.

Remark 3.1 Let $W(s)$ be the optimal squared speed profile along a given path $i_0 = o \to i_1 \to \ldots \to i_{p-1} \to i_p = d$ in $\mathcal{P}_{od}$, and let w_{i_h} be the optimal squared speed at some node i_h of the path, $i_h \neq o, d$. Moreover, let

$$\bar{w}_{i_h} = \min \left\{ \min_{(i_h,j)\in A} \{w^{\max}_{i_h j}, -a^{\min}_{i_h j}\ell_{i_h j}\}, \min_{(k,i_h)\in A} \{w^{\max}_{k i_h}, a^{\max}_{k i_h}\ell_{k i_h}\} \right\}. \tag{4}$$

Then,

1. for each $s \in [s_h, s_{h+1}]$, $W(s) \geq \min\{w_{i_h}, w_{i_{h+1}}\}$;
2. $w_{i_h} \geq \bar{w}_{i_h}$.

According to the previous discussion, the optimal speed profile along a path P from o to d with $|P| + 1$ nodes (here and in what follows $|P|$ denotes the length of path P) is identified once the speeds at nodes of the path are known. Indeed, along any edge (i, j) with given initial and final squared speeds w_i and w_j, the optimal speed profile function is equal to (1) and the traveling time is equal to (2). Then, if we denote by $P(i) \in V$ the node at position i along P, the fastest time $T(P)$ for traversing P is the solution of the following problem:

$$\begin{aligned} T(P) = \min_{\mathbf{w}\in\mathbb{R}^{|P|+1}} & \sum_{i=1}^{|P|} c_{P(i)\,P(i+1)}(w_i, w_{i+1}) \\ & 0 \leq w_i \leq \min\{w^{\max}_{P(i-1),P(i)}, w^{\max}_{P(i),P(i+1)}\}, \quad i \in \{2, \ldots, |P|\}, \\ & w_1 = 0,\ w_{|P|+1} = 0. \end{aligned} \tag{5}$$

We denote by $\mathbf{w}^*(P)$ the optimal solution of this problem and in Appendix B we will describe a recursive procedure to find it, similar to the one employed to define the optimal speed profile function W. According to the discussion above, the problem of finding the speed law which guarantees to traverse a *fixed* path from o to d at a minimum time, under speed and acceleration constraints, is easily solvable even in closed form. But our aim is to compute the minimum time to move from o to d by searching within *all* paths in $\mathcal{P}_{od}$. This is the BASP problem:

$$P^* = \arg\min_{P \in \mathcal{P}_{od}} T(P). \tag{6}$$

As we will see in Section 4, this problem turns out to be NP-hard.

4 Complexity results

In this section we provide two complexity results for BASP. The first result proves NP-hardness of BASP, while the second proves that BASP admits a pseudo-polynomial time algorithm.

4.1 NP-hardness

In this section we prove that, differently from SPP, the BASP variant is NP-hard. We show this by a polynomial reduction of the NP-complete *Partition* problem to BASP. In the Partition problem, given a set $N = \{1, \dots, n\}$ of positive integer values $\beta_1, \dots, \beta_n$, we would like to establish whether N can be partitioned into two subsets N_1 and N_2 such that $\sum_{i \in N_1} \beta_i = \sum_{i \in N_2} \beta_i = \frac{W}{2}$. Given an instance of the Partition problem, we polynomially reduce it to an instance of BASP as follows. Let $G = (V, A)$ be such that:

$$V = N \cup \{0, n+1, n+2\}, \quad A = \{(i,j) \mid i,j \in V \setminus \{n+2\} \wedge i < j\} \cup \{(n+1, n+2)\}.$$

We set the following lengths for the arcs:

$$\ell_{ij} = \begin{cases} 0 & i = 0 \\ \beta_i & i \in \{1, \dots, n\}, \end{cases}$$

while $\ell_{n+1,n+2} = W^2$. For what concerns the maximum speed values, we set $(\forall (i,j) \in A)\ v_{ij}^{\max} = \sqrt{W}$, while we set the maximum acceleration $a_{ij}^{\max} = 1$ and the minimum acceleration $a_{ij}^{\min} = -1$ for all arcs except $(n+1, n+2)$, while we set $a_{n+1,n+2}^{\max} = 0$ and $a_{n+1,n+2}^{\min} = -\frac{1}{2W}$. Note that, according to the imposed restrictions, $a_{n+1,n+2}^{\max}$ should be strictly larger than 0. However, the result proved with null maximum acceleration can be extended, by continuity, to any sufficiently small and positive maximum acceleration. The origin node o is node 0, with zero speed, while the

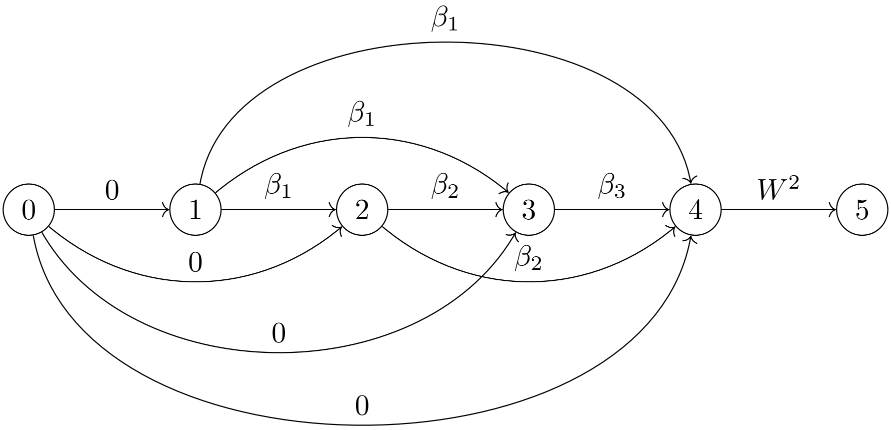


Fig. 4 An instance of BASP derived from the Partition problem for $n = 3$. Along each arc the maximum allowed speed is $\sqrt{W}$ ($W = \beta_1 + \beta_2 + \beta_3$), $a^{\max}$ is equal to 1 and $a^{\min}$ is equal to -1 along all arcs except (4, 5) where $a^{\max}_{45} = 0$ and $a^{\min}_{45} = -\frac{1}{2W}$. Finally, $v_0 = v_5 = 0$

destination node d is $n + 2$, with zero speed. An example of BASP instance derived from the Partition problem with $n = 3$ is illustrated in Fig. 4.

We prove the following.

Proposition 4.1 *The optimal value of the BASP instance introduced above is equal to* $\sqrt{W} + 2W^{\frac{3}{2}}$ *if and only if the partition problem admits a solution and is otherwise larger than such value.*

The proof of Proposition 4.1 is presented in Appendix A.

Remark 4.1 The complexity result given above shows that BASP is NP-hard even in case all arcs except one share the same acceleration and deceleration bounds. It is still an open question whether NP-hardness still holds if all arcs have the same bound. However, in Sect. 5.1 we will show that optimal solutions for this case are elementary paths (paths with no node repetition). This allows to derive sharper approximation results.

4.2 Pseudo-polynomial algorithm

Although BASP is NP-complete, the following proposition shows it admits a pseudo-polynomial algorithm under the assumption of integer data.

Proposition 4.2 *Let us assume that all problem data,* ℓ_{ij}, $v^{\max}_{ij}$, $a^{\max}_{ij}$, *and* $a^{\min}_{ij}$ *for all* $(i, j) \in A$ *are integer values. Then, BASP admits a pseudo-polynomial algorithm.*

Proof First, we observe that at optimal solutions there is a finite number of speeds which can be reached at each node and the squares of such speeds are integer values. Indeed, the squared speed at some node i is:

- either equal to $w_{ij}^{\max}$, for some j such that $(i,j) \in A$, which is an integer value by assumption;
- or equal to $w_{ki}^{\max}$, for some k such that $(k, i) \in A$, which is again an integer value by assumption;
- or i is the end point of a sub-path $j_0 \to j_1 \to \cdots j_{k-1} \to j_k = i$, with squared speed at node j_1 equal to $w_{j_0 j_1}^{\max}$ and squared speed at node i

  $$w_i = w_{j_0 j_1}^{\max} + 2 \sum_{h=1}^{k-1} a_{j_h j_{h+1}}^{\max} \ell_{j_h j_{h+1}},$$

  which is an integer value due to integrality of all the data (see Fig. 5). Note that j_1 may be the starting node o, in which case $w_{j_1} = 0$ (j_0 is not included in this case);
- or i is the starting point of a path $j_0 = i \to j_1 \to \cdots j_{k-1} \to j_k$, with squared speed at node j_{k-1} equal to $w_{j_{k-1} j_k}^{\max}$ and squared speed at node i

  $$w_i = w_{j_{k-1} j_k}^{\max} - 2 \sum_{h=0}^{k-2} a_{j_h j_{h+1}}^{\min} \ell_{j_h j_{h+1}},$$

  which is, again, an integer value due to integrality of all the data (see Fig. 6). Note that node j_{k-1} may be the destination node d, in which case $w_{j_{k-1}} = 0$ (j_k is not included in this case).

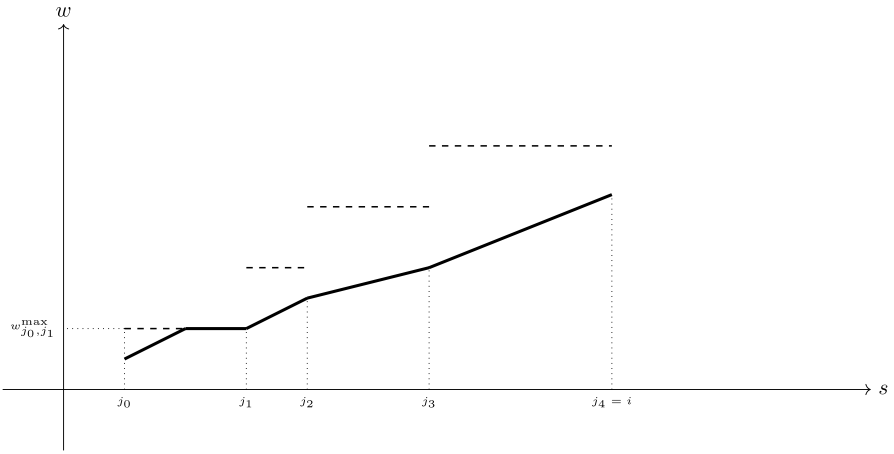


Fig. 5 Optimal (squared) speed profile from node j_0 up to node i (continuous line) when node i is reached by accelerating as much as possible along all arcs between j_1 and i. The dashed lines represent the maximum squared speeds along the arcs

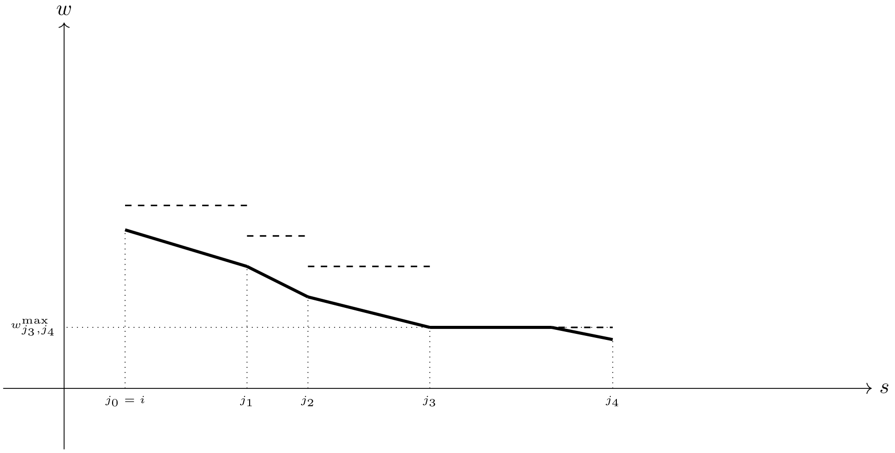


Fig. 6 Optimal (squared) speed profile from node i up to node j_4 (continuous line) when node j_4 is reached by decelerating as much as possible along all arcs between i and j_4. The dashed lines represent the maximum squared speeds along the arcs

Thus, the set $\mathcal{W}$ of different possible squared speeds can be taken equal to the set of all integers between 0 and $\bar{W} = \max_{(i,j)\in A} w^{\max}_{ij}$. Now we create a new graph with node set $V \times \mathcal{W}$, that is, each node is a pair made up by a node in V and one of the possible squared speeds in $\mathcal{W}$. Thus, the number of nodes is $\bar{W}|V|$. For what concerns the arc set, in this graph an arc between node (i, w_i) and node (j, w_j) exists if there exists an arc $(i,j) \in A$ and, moreover, $c_{ij}(w_i, w_j) < +\infty$, that is, there exists a feasible speed profile along arc (i, j) with initial squared speed w_i and final squared speed w_j. Then, the number of arcs is limited from above by $\bar{W}^2|A|$. The distance associated to this arc is the minimum time for a path from i to j with the boundary conditions w_i and w_j, which can be easily computed through (2), as discussed in Sect. 3 (recall that, in case w_i and w_j are not feasible, as illustrated in Fig. 2b, then the arc is removed). Then, we can solve our problem by applying, for instance, Dijkstra algorithm to this graph. Dijkstra's complexity is $O(m + n\log(n))$ and is, thus, polynomial with respect to the size and the data of the original problem, which proves pseudo-polynomiality. □

Remark 4.2 While Proposition 4.2 has been proved under the assumption of integer data, it can also be extended to rational data. In such case the squared speeds which can be reached by optimal solutions are not integer values but are multiple of a rational number $\frac{1}{t}$, where t depends on the problem data. Of course, the size of the extended graph increases with t. The approximation algorithm discussed in the following Sect. 5 is motivated by the need to consider a discretization step larger than $\frac{1}{t}$ in order to have a graph of manageable size.

5 Approximation algorithm

In this section we present an approximation algorithm for BASP with a complexity that is polynomial with respect to the size and the data of the original problem and the inverse of the approximation factor. The idea is to discretize the squared speeds in order to obtain a finite set of possible squared speeds at each node of the graph. In this way, imposing that the initial and final squared speeds along each arc belong to the discretized set of squared speeds, the set of possible speed profiles over each arc becomes finite. Hence, we can define an extended graph that enables us to solve this discretized version of the problem by means of Dijkstra's algorithm. Differently from Proposition 4.2, here arc lengths, accelerations and speed bounds need not be integer values (actually, the approach could also be extended to the case of non-constant speed bounds along arcs as discussed in Remark 2.1). In this case, we just discretize the squared speeds and impose the additional constraint that the squared speeds at the beginning and at the end of each arc belong to the set of discretized squared speeds. Let

$$\Omega_h = \{\omega \in [0, \bar{W}] \mid (\exists k \in \mathbb{N})\ \omega = kh\},$$

with $\bar{W} = \max_{(i,j)\in A} w_{ij}^{\max}$, be the set of discretized squared speeds with discretization step h. Then, $|\Omega_h| = \lceil \bar{W}/h \rceil$. The discretized problem is defined over a graph that extends graph G of the original problem. Namely, the extended graph $G' = (V', A')$ is defined as follows:

$$V' = \left\{ (i, \omega) \mid i \in V\ \wedge\ \omega \in \Omega_h\ \wedge\ \omega \leq \max\left\{ \max_{j \mid (i,j)\in A} w_{ij}^{\max}, \max_{k \mid (k,i)\in A} w_{ki}^{\max} \right\} \right\},$$

where we notice that we bound from above the possible squared speeds at node i by the maximum squared speeds along the incoming and outgoing arcs of node i, while

$$A' = \{((i, \omega_i), (j, \omega_j)) \in V' \times V' \mid (i,j) \in A\ \wedge\ \omega_i, \omega_j \leq w_{ij}^{\max}\ \wedge\ c_{ij}(\omega_i, \omega_j) < +\infty\},$$

where we recall that $c_{ij}(\omega_i, \omega_j) = +\infty$ means that no feasible profile is able to travel from i to j with initial and final squared speeds equal to ω_i and ω_j, respectively, while $c_{ij}(\omega_i, \omega_j) < +\infty$, as defined in (2), is the optimal travel time along arc (i, j) with the given initial and final squared speeds.

Remark 5.1 The cost for constructing the extended graph G' is $O(|A| \cdot |\Omega_h|^2)$. Indeed, the number of arcs of the extended graph $|A'|$ is bounded from above by $|A| \cdot |\Omega_h|^2$ and the cost for checking whether an arc exists or not in the extended graph, that is the cost for checking conditions (3), is constant.

Once the extended graph has been defined, the proposed approximation algorithm is nothing but the application of Dijkstra's algorithm to solve the discretized problem. More precisely, we search for the shortest path connecting nodes $(o, 0), (d, 0) \in V'$ over graph G', where the cost of arc $((i, \omega_i), (j, \omega_j)) \in A'$ is equal to the value $c_{ij}(\omega_i, \omega_j) < +\infty$ defined in (2). Then, we have the following complexity result for the approximation algorithm.

Proposition 5.1 *The complexity of the approximation algorithm is*

$$O\left(m\left(\frac{\bar{W}}{h}\right)^2 + \frac{n\bar{W}}{h}\log\left(\frac{n\bar{W}}{h}\right)\right).$$

Proof The approximation algorithm is Dijkstra's algorithm applied on the extended graph G', so that its complexity is $O(|A'| + |V'|\log|V'|)$. Then, the result immediately follows by observing that $|V'| \leq |V||\Omega_h| = n\lceil\bar{W}/h\rceil$ and $|A'| \leq |A||\Omega_h|^2 \leq m\left(\lceil\bar{W}/h\rceil\right)^2$. □

As a next step, we want to obtain an estimate of the absolute error in terms of travel time of the discretized solution returned by the approximation algorithm with respect to the continuous solution of the original BASP problem. To this end, let us consider the optimal path $P^* \in \mathcal{P}_{od}$

$$o \to i_1 \to i_2 \to \cdots \to d,$$

of the original BASP problem, with the corresponding squared speeds at each node $\{w_o = 0, w_{i_1}, w_{i_2}, \ldots, w_d = 0\}$. Our aim is to build a feasible solution of the discretized problem traversing the same arcs as path P^* and whose speed profile is not above the optimal speed profile of the original BASP problem but is *as close as possible* to it. Building such solution requires some care. It is tempting to proceed as follows: for each node i in the optimal path P^*, with squared speed w_i in the optimal speed profile of BASP, replace w_i with

$$\omega_i = \max\{kh \mid kh \leq w_i,\ k \in \mathbb{N}\}, \tag{7}$$

that is, with the largest discretized speed bounding from below w_i. Unfortunately, this does not work. Indeed, let us consider some arc $(i,j) \in P^*$ with the related optimal squared speeds w_i and w_j, and let ω_i and ω_j be chosen as in (7). Unfortunately, in the extended graph, arc $((i,\omega_i),(j,\omega_j))$ may not exist, since a situation like the one displayed in Fig. 2b may occur. Formally, according to (3), it may happen that either $\omega_i + 2a_{ij}^{\max}\ell_{ij} < \omega_j$ or $\omega_j - 2a_{ij}^{\min}\ell_{ij} < \omega_i$. Therefore, we need to proceed differently to build a solution of the discretized problem starting from the optimal solution of BASP. For $x, h \in \mathbb{R}$, $h > 0$, we denote by $\langle x\rangle = \max\{i \in \mathbb{Z} \mid ih \leq x\}$ the maximum multiple of h lower than or equal to x. First, we reformulate the discretized problem as follows. Let P be an assigned path from o to d. The minimum time $T^h(P)$ to traverse P in the discretized problem is:

$$
\begin{aligned}
T^h(P) = \min_{\mathbf{w}\in\mathbb{R}^{|P|+1}} & \sum_{i=1}^{|P|} c_{P(i)\,P(i+1)}(w_i, w_{i+1}) \\
& 0 \le w_i \le \min\{w^{\max}_{P(i-1),P(i)}, w^{\max}_{P(i),P(i+1)}\}, \quad i \in \{2,\dots,|P|\}, \\
& w_1 = 0,\ w_{|P|+1} = 0\,, \\
& \langle w_i \rangle = w_i\,, \quad i \in \{1,\dots,|P|+1\}.
\end{aligned}
\tag{8}
$$

Problem (8) is obtained by adding to Problem (5) the last constraint, imposing that the coordinates of $\mathbf{w}$ have to be multiples of h. We call $\mathbf{w}^h(P) \in \mathbb{R}^{|P|+1}$ a vector that corresponds to the solution of (8). Note that Problem (8) always admits at least one feasible solution with finite objective function value, for instance the solution $w_i = 0$ for all $i \in \{1,\dots,|P|+1\}$. The solution of discretized BASP corresponds to path

$$
P^h = \arg\min_{P\in\mathcal{P}_{od}} T^h(P). \tag{9}
$$

Recall that this problem can be solved by the application of Dijkstra's algorithm to the extended graph described above. The following lemma compares the optimal values and solutions of Problems (5) and (8).

Lemma 5.1 *For any path P and any $h > 0$,*

(i) $T^h(P) \ge T(P)$,
(ii) $\mathbf{w}^h(P) \le \mathbf{w}^*(P)$ (*vector inequalities are intended component-wise*).

Proof Statement (i) is a consequence of the fact that Problem (8) is obtained by adding a constraint to Problem (5). For Statement (ii), first note that, if $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{|P|+1}$ are feasible values for $\mathbf{w}$ in Problem (5), then also their component-wise maximum $\mathbf{w} = \max\{\mathbf{w}_1, \mathbf{w}_2\}$ is feasible (see, for instance, [4]). By contradiction, if $\mathbf{w}^h(P) \nleq \mathbf{w}^*(P)$ then $\mathbf{w} = \max\{\mathbf{w}^h(P), \mathbf{w}^*(P)\}$ is feasible for Problem (5). Note that the objective function $f(\mathbf{w}) = \sum_{i=1}^{|P|} c_{P(i)\,P(i+1)}(w_i, w_{i+1})$ of Problem (5) is strictly decreasing, that is, if $\mathbf{w}_1 \ge \mathbf{w}_2$ and $\mathbf{w}_1 \ne \mathbf{w}_2$, then $f(\mathbf{w}_1) < f(\mathbf{w}_2)$. Since $\mathbf{w} \ge \mathbf{w}^*(P)$ and $\mathbf{w} \ne \mathbf{w}^*(P)$, it follows that $f(\mathbf{w}) < f(\mathbf{w}^*(P))$, contradicting the optimality of $\mathbf{w}^*(P)$. □

Next, the following lemma, whose proof is given in Appendix B, gives an upper bound on the components of the difference vector $\mathbf{w}^*(P) - \mathbf{w}^h(P)$. Note that such components are nonnegative in view of part ii) of Lemma 5.1.

Lemma 5.2 *The following holds for all $i \in \{1,\dots,|P|+1\}$:*

$$
w^*(P)_i - w^h(P)_i \le h|P|.
$$

In order to use Lemma 5.2, we need to find an upper bound on $|P^*|$, the number of arcs of the optimal path. This can be done as follows.

Lemma 5.3 *An upper bound for the number of arcs $|P^*|$ of the optimal path P^* is given by*

$$|P^*| \leq \left(\frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + 2\frac{\sqrt{\bar{w}}}{a_{\min}} \right) \frac{\sqrt{\bar{W}}}{\ell_{\min}}, \tag{10}$$

where

- P_{SPP} is the shortest path connecting o to d over the original graph, when the cost of each arc $(i,j) \in A$ is equal to ℓ_{ij};
- $a_{\min} = \min_{(i,j)\in A} \min\{a_{ij}^{\max}, -a_{ij}^{\min}\} > 0$;
- $\bar{w} = \min_{i \in V\setminus\{o,d\}} \bar{w}_i$, where $\bar{w}_i > 0$ is defined in (4);
- $\bar{W} = \max_{(i,j)\in A} w_{ij}^{\max}$;
- $\ell_{\min} = \min_{(i,j)\in A} \ell_{ij}$.

Proof A feasible solution for BASP is obtained by the arcs of path P_{SPP}, along which, starting from null speed at node o, we first accelerate with acceleration $a_{\min}$ until we reach speed $\bar{w}$, then we keep the speed constant and, finally, we decelerate with deceleration $a_{\min}$ until we reach the final null speed at node d. The time t_{SPP} to travel along path P_{SPP} with the given speed profile, is an upper bound for the travel time of the optimal path P^*. We have that

$$t_{\mathrm{SPP}} \leq \frac{\sqrt{\bar{w}}}{a_{\min}} + \frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + \frac{\sqrt{\bar{w}}}{a_{\min}}.$$

Next we need a lower bound for the time needed to travel along any arc $(i,j) \in A$. We denote this time with $t_{\min}$ and a lower bound for it is $\ell_{\min}/\sqrt{\bar{W}}$. Then, the ratio of t_{SPP} to $t_{\min}$ is an upper bound for the number of arcs $|P^*|$ in the optimal path, that is,

$$|P^*| \leq \frac{t_{\mathrm{SPP}}}{t_{\min}} \leq \left(\frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + 2\frac{\sqrt{\bar{w}}}{a_{\min}} \right) \frac{\sqrt{\bar{W}}}{\ell_{\min}}.$$

□

In the following, we estimate the difference $T^h(P^h) - T(P^*) \geq 0$ between the optimal values of the discretized BASP (9) and BASP (6) (nonnegativity follows form part i) of Lemma 5.1). Path P^h, corresponding to the solution of the discretized BASP, can be different from P^* and $T^h(P^h) \leq T^h(P^*)$. Hence, $T^h(P^h) - T(P^*) \leq T^h(P^*) - T(P^*)$. Quantities $T(P^*)$ and $T^h(P^*)$ correspond to the optimal values of Problems (5) and (8) on the same path P^*. In order to bound the difference $T^h(P^*) - T(P^*)$, we need a further lemma, giving, for some arc (i, j), an upper bound for the difference $c_{ij}(w, z) - c_{ij}(\hat{w}, \hat{z})$ (that is, the time difference between the times needed to travel arc (i, j) with boundary speeds w, z and $\hat{w}, \hat{z}$, respectively, with $w \leq \hat{w}$ and $z \leq \hat{z}$). The lemma will be proved in Appendix C.

Lemma 5.4 *For any arc (i, j) and any $w, z, \hat{w}, \hat{z} \in \mathbb{R}$, with $0 \leq w, z, \hat{w}, \hat{z} \leq w_{ij}^{\max}$, $w \leq \hat{w}$, $z \leq \hat{z}$, and $c_{ij}(w, z), c_{ij}(\hat{w}, \hat{z}) < +\infty$, the following bound holds*:

$$\left| c_{ij}(w, z) - c_{ij}(\hat{w}, \hat{z}) \right| \leq \frac{4 \max\{|w - \hat{w}|, |z - \hat{z}|\}}{a_{\min} \sqrt{\bar{w}}},$$

where $a_{\min}$ and $\bar{w}$ are defined in the statement of Lemma 5.3.

Then, we can provide an estimate on $T^h(P^*) - T(P^*)$ (which also bounds $T^h(P^h) - T(P^*)$) by summing up the contributions of all arcs of P^*.

Proposition 5.2 *The following bound holds*:

$$T^h(P^h) - T(P^*) \leq \left(\frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + 2\frac{\sqrt{\bar{w}}}{a_{\min}} \right)^2 \frac{4\bar{W}}{\ell_{\min}^2 a_{\min} \sqrt{\bar{w}}} h,$$

where P_{SPP} is the shortest path connecting node o to node d.

Proof First observe that, for any arc $e = (P^*(i), P^*(i+1))$, it holds that

$$\begin{aligned} &\left| c_e(w_i^h(P^*), w_{i+1}^h(P^*)) - c_e(w_i^*(P^*), w_{i+1}^*(P^*)) \right| \\ &\quad \leq \frac{4 \max\{|w_i^h(P^*) - w_i^*(P^*)|, |w_{i+1}^h(P^*) - w_{i+1}^*(P^*)|\}}{a_{\min} \sqrt{\bar{w}}} \leq \frac{4|P^*|h}{a_{\min} \sqrt{\bar{w}}}, \end{aligned} \tag{11}$$

as a consequence of Lemma 5.4 (first inequality) and of Lemma 5.2 (second inequality). Then, it follows that

$$\begin{aligned} T^h(P^h) - T^*(P^*) \leq & T^h(P^*) - T^*(P^*) \leq \sum_{k=1}^{|P^*|} \frac{4|P^*|h}{a_{\min} \sqrt{\bar{w}}} \\ = & \frac{4|P^*|^2 h}{a_{\min} \sqrt{\bar{w}}} \leq \left(\frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + 2\frac{\sqrt{\bar{w}}}{a_{\min}} \right)^2 \frac{4\bar{W}}{\ell_{\min}^2 a_{\min} \sqrt{\bar{w}}} h, \end{aligned}$$

where the first inequality derives from (11), whilst the second one follows from Lemma 5.3. □

We can also provide an estimate of the relative error.

Proposition 5.3 *Given $\epsilon \in (0, 1)$, the relative error of the approximated problem with $h = C\epsilon$, where C is a constant that depends on the problem data, is $1 + \epsilon$, that is, $\frac{T^h(P^h)}{T(P^*)} \leq 1 + \epsilon$.*

Proof Let $t_{\min}$ be the travel time of the arc of shortest length $\ell_{\min}$ assuming that the squared speed along it is constantly equal to $\bar{W}$ (this is a lower bound for the travel time along any arc). Then, by Proposition 5.2, we have the following estimate over the relative error:

$$\frac{T^h(P^h)}{T(P^*)} \leq 1 + \frac{T(P^*) - T^h(P^h)}{T(P^*)} \leq 1 + \frac{T(P^*) - T^h(P^h)}{t_{\min}} = 1 + (T(P^*) - T^h(P^h))\frac{\sqrt{\bar{W}}}{\ell_{\min}}$$
$$\leq 1 + \left(\frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + 2\frac{\sqrt{\bar{w}}}{a_{\min}}\right)^2 \frac{3\bar{W}^{\frac{3}{2}}}{\ell^3_{\min} a_{\min}\sqrt{\bar{w}}} h = 1 + \epsilon,$$

with

$$h = C\epsilon, \qquad C = \left[\left(\frac{\ell(P_{\mathrm{SPP}})}{\sqrt{\bar{w}}} + 2\frac{\sqrt{\bar{w}}}{a_{\min}}\right)^2 \frac{3\bar{W}^{\frac{3}{2}}}{\ell^3_{\min} a_{\min}\sqrt{\bar{w}}}\right]^{-1}. \tag{12}$$

□

The following theorem states a time-complexity and an error estimate result for the approximated problem.

Theorem 5.1 *Given $\epsilon \in (0,1)$, let $h = C\epsilon$ with C defined as in (12). Then, $\frac{T^h(P^h)}{T(P^*)} \leq 1 + \epsilon$, that is the solution returned by the approximation algorithm has relative error at most ϵ, and the approximation algorithm has time-complexity*

$$O\left(\left(\frac{\bar{W}}{C\epsilon}\right)^2 m + \frac{\bar{W}n}{C\epsilon}\log\left(\frac{\bar{W}n}{C\epsilon}\right)\right).$$

Proof The thesis directly follows from Propositions 5.1 and 5.3. □

5.1 The case of uniform acceleration bounds

As previously mentioned in Remark 4.1, it is still unclear whether the case where all arcs share the same acceleration and deceleration bounds, denoted with $a^{\max}$

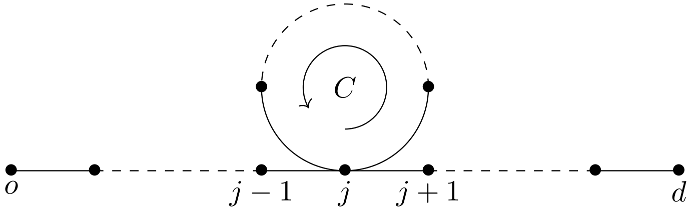


Fig. 7 Two distinct paths from node o to node d: π and $\pi' = C \cup \pi$

and $a^{\min}$, respectively, is NP-hard or not. However, we can prove the following result.

Proposition 5.4 *If*

$$(\forall (i,j) \in A)\ a_{ij}^{\max} = a^{\max}\ \wedge\ a_{ij}^{\min} = a^{\min},$$

that is, all arcs have the same acceleration and deceleration bounds, then the optimal solution of BASP is an elementary path (a path that does not contain loops).

Proof Let us consider two distinct paths π and π' from node o to node d such that π is elementary and $\pi \subset \pi'$. Since π' contains the entire path π, it must contain a cycle C, as shown in Fig. 7.

We will show that the time needed to traverse π is lower than the time needed to traverse π', so that we can restrict the attention to elementary paths. Let j be the first node of cycle C in π' as indicated in Fig. 7. Let v_j be the initial speed at node j along path π. Along cycle C we can speed up. If, after traveling cycle C, the new speed at node j is $v_C > v_j$, then there is a time gain $\tilde{t}$ in traveling the final sub-path $j \to d$, since the new initial speed at j is higher. However, there is also a loss of time $\bar{t}$ to travel along cycle C, We prove that the latter is always greater than the former. Indeed, a lower bound $\bar{t}_{\min}$ for the time needed to travel along cycle C is obtained by assuming that the maximum acceleration $a^{\max}$ can be kept along the cycle without hitting maximum speed bounds:

$$\bar{t}_{\min} = \frac{v_C - v_j}{a^{\max}}.$$

Now, let ℓ be the length of the sub-path $j \to d$. An upper bound $\Delta t_{\max}$ on the time difference between the time to travel along the sub-path $j \to d$ with starting speed v_j and the time to travel along the same sub-path with starting speed v_C is obtained by assuming that we can keep the maximum acceleration along the sub-path:

$$\Delta t_{\max} = \frac{\sqrt{v_j^2 + \ell a^{\max}} - v_j}{a^{\max}} - \frac{\sqrt{v_C^2 + \ell a^{\max}} - v_C}{a^{\max}}.$$

It follows that

$$\bar{t}_{\min} > \Delta t_{\max} \iff \sqrt{v_C^2 + \ell a^{\max}} > \sqrt{v_j^2 + \ell a^{\max}},$$

which holds true in view of $v_C > v_j$.

The same reasoning applies to maximum deceleration $a^{\min}$. Indeed, suppose that along path π the agent has to decelerate before j to reach the speed v_j. In this case, along cycle C we could speed down, allowing to maintain a higher speed along the sub-path $o \to j$. However, it is possible to prove that the lost time $\bar{t}$ to travel along cycle C is always larger than the time gain $\tilde{t}$ in traveling the initial sub-path $o \to j$ with a higher final speed, and so π is faster than π' in any case. □

This result is important because it allows us to state that in this sub-case the upper bound for the number of arcs of the optimal path P^* depends only on the number of nodes n. In particular

$$|P^*| \leq n - 1.$$

Therefore, we can use this bound rather than the one derived in Lemma 5.3 in all subsequent results. So, for instance, the upper bound in Proposition 5.2 can be set equal to $\frac{4n^2h}{a_{\min}\sqrt{\bar{w}}}$.

6 Computational experiments

In this section we test the approximation algorithm introduced in Sect. 5. We consider two real-life industrial scenarios of warehouses. The problem data have been provided by packaging company Ocme S.r.l., based in Parma, Italy. The simulations have been performed on a 2.7 GHz Intel Core i5 dual-core with 8GB of RAM. The acceleration and deceleration bounds on both scenarios are given by $\alpha^+ = 0.28$ m s^{-2} and $\alpha^- = -0.18$ m s^{-2}, for arcs with mean curvature smaller than or equal to 0.25. Whilst on arcs whose mean curvature is larger than this value, the bounds are given by $\alpha^+ = 0.14$ m s^{-2} and $\alpha^- = -0.09$ m s^{-2}. The first scenario is modeled as a graph $G = (V, A)$ with 368 nodes and 679 edges and is depicted in Fig. 8.

The speed bounds on both scenarios are constant for each arc but vary from arc to arc, according to the associated paths curvatures. For the first scenario they take values in interval [0.136, 1.7] m s^{-1}, whilst the arc-lengths take values between 0.628 and 10.87 m and have an average value of 2.863 m.

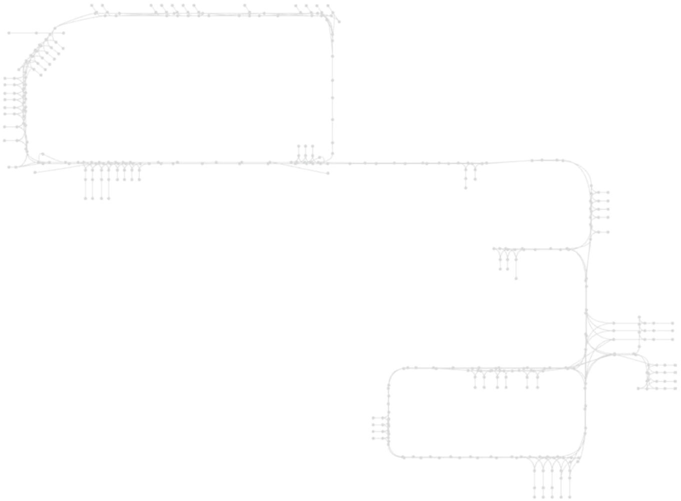


Fig. 8 First real-life industrial scenario

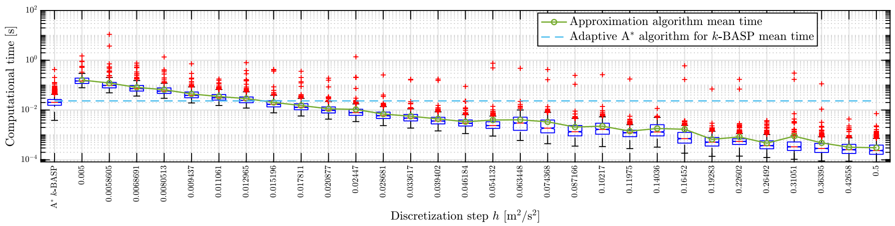


Fig. 9 Approximation algorithm computational times for different values of h on the first scenario

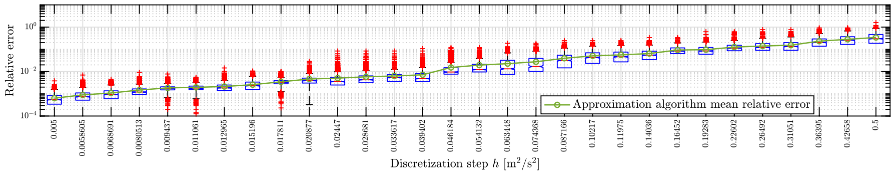


Fig. 10 Approximation algorithm relative error for different values of h on the first scenario

In what follows, we compare the approximation algorithm of Sect. 5 with the adaptive A* algorithm for k-BASP presented in [1]. Note that the adaptive algorithm solves a k-BASP instance in polynomial time complexity with respect to the number of edges and vertices of the associated graph, but its complexity is exponential with respect to k. In the following, the adaptive A* algorithm for k-BASP is initialized with $k = 3$. Again, we refer the reader to Ardizzoni et al. [1] for a more in depth discussion.

Given the graphs associated to the considered automated warehouses, we generated the extended graphs associated to them for different values of the discretization step h of the squared speeds. For the first scenario, step h takes values in a set H of thirty logarithmically spaced values between 0.005 and 0.5 m^2 s^{-2}, hence, we considered thirty different sets of discretized squared speeds Ω_h and extended graphs $G'_h = (V'_h, A'_h)$. We generated 1000 random pairs of source-target nodes $\{(s_i, t_i)\}_{i\in\{1,\dots,1000\}}$ in $V \times V$. Then, for each of the previous pairs of source-target nodes, we considered the corresponding pair $(s_i^h, t_i^h) = ((s_i, 0), (t_i, 0)) \in V'_h \times V'_h$ on extended graph G'_h and ran Dijkstra's algorithm on G'_h in order to obtain a trajectory starting at node s_i with null speed and ending at node t_i with null speed, for $h \in H$. Figure 9 shows the box-and-whisker plot of the computational times of Dijkstra's algorithm for different values of the discretization step h for solving the 1000 randomly generated instances and compares them with those of the adaptive A* algorithm for k-BASP on the same set of instances.

Note that, as the discretization step h increases, the number of discretized squared speeds decreases, hence, the number of nodes and edges in graph G'_h decreases as well, making Dijkstra's algorithm explore a smaller graph and run faster. Also, observe that, for values of h greater than 0.015159 m^2 s^{-2}, the mean computational times of Dijkstra's algorithm on extended graphs G'_h (represented by a green line with circles) are better than that of the adaptive A* algorithm for k-BASP (represented by

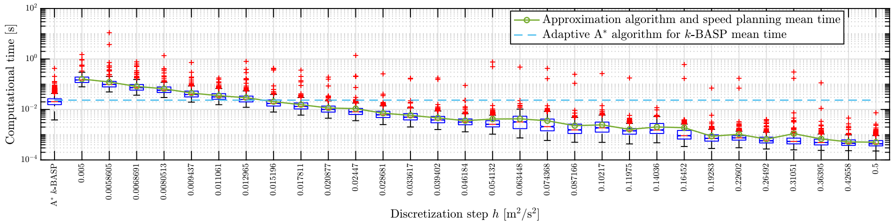


Fig. 11 Approximation algorithm and speed planning algorithm computational times for different values of h on the first scenario

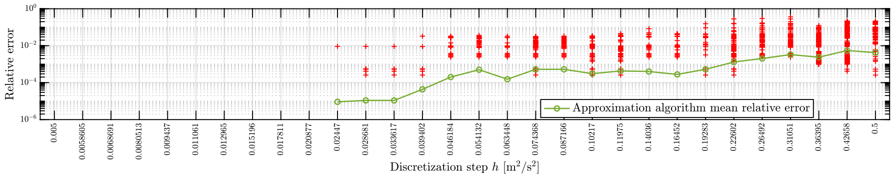


Fig. 12 Approximation algorithm and speed planning algorithm relative error for different values of h on the first scenario

a dashed line). On the other hand, as the discretization step h increases, so does the relative error on the travel time, which, for values of $h \geq 0.19283$ m^2 s^{-2}, is larger than 10^{-1}. Figure 10 represents the box-and-whisker plot of the relative error. Note that we set a tolerance on the relative error of the trajectories obtained with the approximation algorithm, relative errors smaller than 10^{-4} are not considered. This roughly corresponds to an absolute error of the order of 10^{-2} s.

A good compromise for this scenario could be $h = 0.063448$ m^2 s^{-2} which is associated to a mean computational time that is 5.76 times faster than that of the adaptive A* algorithm for k-BASP, while at the same time maintaining a mean relative error of the order of $2 \cdot 10^{-2}$. We could also exploit the approximation algorithm just for obtaining a path and then compute the optimal speed profile along such a path by the procedure described in Sect. 3. In this way we could employ a bigger discretization step h for achieving small computational times while maintaining high precision. The speed planning algorithm described in Sect. 3 has linear-time computational complexity with respect to the number of nodes of the path. Figure 11 shows the box-and-whisker plot of the computational times of the approximation algorithm, as in Fig. 9, summed with those of the speed planning algorithm applied on the obtained paths for the 1000 randomly generated instances on the first scenario.

Figure 12 shows the box-and-whisker plot of the relative error on the travel time of the trajectories obtained coupling the approximation algorithm of Sect. 5 with the speed planning one. Again, we set a tolerance on relative errors of 10^{-4}. In this case the mean relative errors are on average two orders of magnitude smaller than those presented in Fig. 10 and the percentage of solutions with a relative error smaller than 10^{-4} ranges from 93.1% with $h = 0.5$ m^2 s^{-2} to 100% for $h = 0.005$ m^2 s^{-2}.

For this scenario, if we fix $h = 0.5$ m^2 s^{-2}, we get a mean computational time that is 46.35 times faster than that of the adaptive A* algorithm for k-BASP, while obtaining a solution with a mean relative error of 4×10^{-3}, which is one order of magnitude smaller than that of the approximation algorithm alone with $h = 0.063448$ m^2 s^{-2}.

It is worthwhile to remark that there always exists a sufficiently small value h such that the optimal path of the discretized problem coincides with the optimal path of the continuous problem. Indeed, if h is chosen in such a way that the absolute error of the approximation algorithm, bounded from above as discussed in Proposition 5.2, is lower than the difference between the traveling times of the best and second-best path, then the approximation algorithm returns the best path (that is, according to the notation introduced in Sect. 5, P^h is equal to P^*). However, the difference is not known in advance and even in case it were known, the choice of h based on the upper bound stated in Proposition 5.2 may lead to an excessively small value.

The second scenario is modeled as a graph with 2419 nodes and 4255 edges and is depicted in Fig. 13.

For this scenario the speed bounds take values in interval [0.1, 1.7] m s^{-1}, whilst the arc-lengths take values between 0.2 and 18.15 m and have an average value of 4.24 m. For the first scenario, step h takes values in a set H of ten

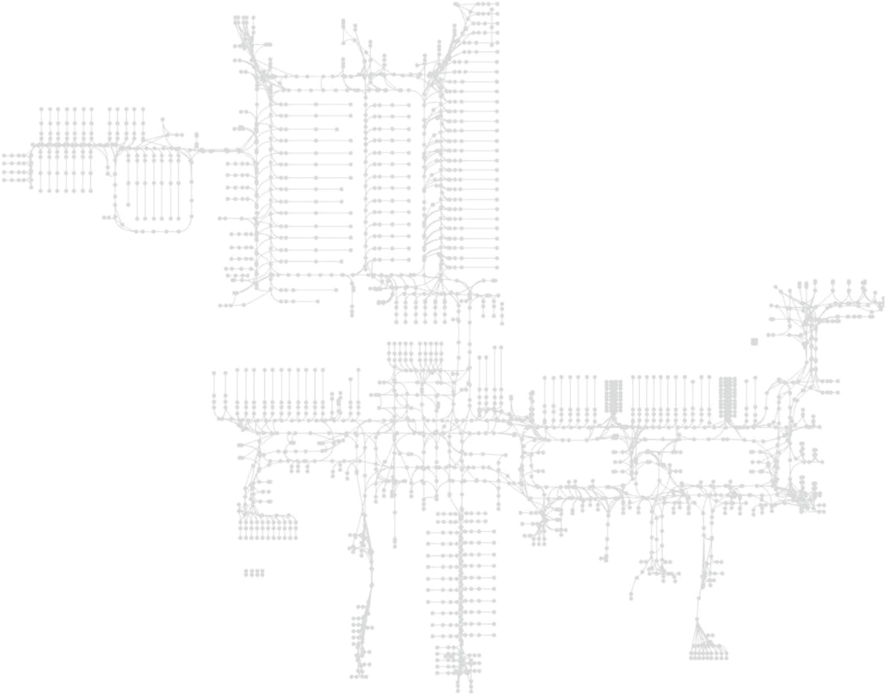


Fig. 13 Second real-life industrial scenario

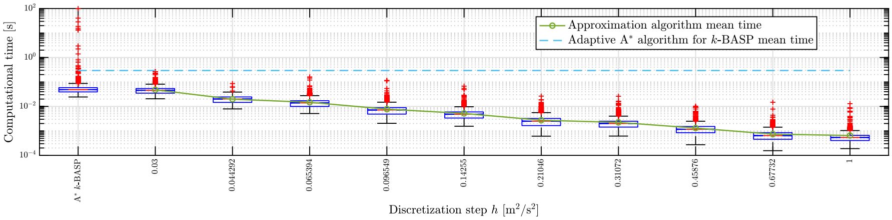


Fig. 14 Approximation algorithm computational times for different values of h on the second scenario

logarithmically spaced values between 0.03 and 1 $\mathrm{m}^2\,\mathrm{s}^{-2}$, hence, we considered ten different sets of discretized squared speeds Ω_h and extended graphs $G'_h = (V'_h, A'_h)$. As for the previous scenario, we generated 1000 random pairs of source-target nodes $\{(s_i, t_i)\}_{i \in \{1,\ldots,1000\}}$ in $V \times V$ and tested the approximation algorithm on such pairs. Figure 14 shows the box-and-whisker plot of the computational times of Dijkstra's algorithm for different values of the discretization step h for solving the 1000 randomly generated instances and compares them with those of the adaptive A* algorithm for k-BASP on the same set of instances.

Observe that, for all values of h, the mean computational times of Dijkstra's algorithm on extended graphs G'_h (represented by a green line with circles) are better than that of the adaptive A* algorithm for k-BASP (represented by a dashed line). However, note that the median computational time of the latter is almost the same as that of the approximation algorithm with $h = 0.03\,\mathrm{m}^2\,\mathrm{s}^{-2}$ (both represented as horizontal red lines within their corresponding blue boxes). This is due to the fact that the adaptive A* algorithm for k-BASP presents a small group of outliers with very high computational times compared to its median. On the other hand, as the discretization step h increases, so does the relative error on the travel time, which, for values of $h \geq 0.45876\,\mathrm{m}^2\,\mathrm{s}^{-2}$, is larger than 10^{-1}. Figure 15 represents the box-and-whisker plot of the relative error for which we set a tolerance of 10^{-4}.

A good compromise for this scenario could be $h = 0.21046\,\mathrm{m}^2\,\mathrm{s}^{-2}$ which is associated to a mean computational time that is 107.3 times faster than that of the adaptive A* algorithm for k-BASP, while at the same time maintaining a mean relative error of the order of $3 \cdot 10^{-2}$. Figure 16 shows the box-and-whisker plot of the computational times of the approximation algorithm, as in Fig. 14, summed with those of the speed planning algorithm described in Sect. 3 applied on the obtained paths for the 1000 randomly generated instances on the first scenario.

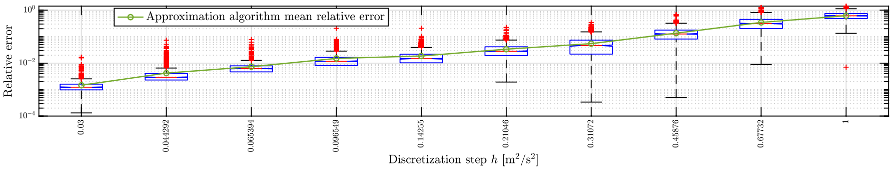


Fig. 15 Approximation algorithm relative error for different values of h on the second scenario

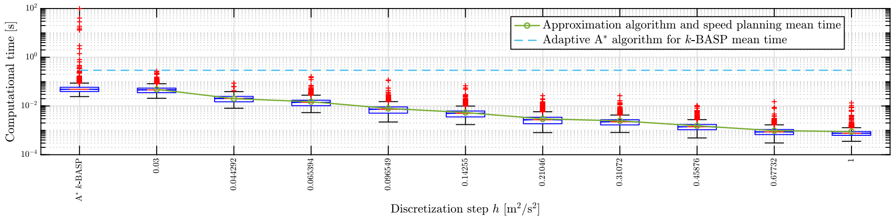


Fig. 16 Approximation algorithm and speed planning algorithm computational times for different values of *h* on the second scenario

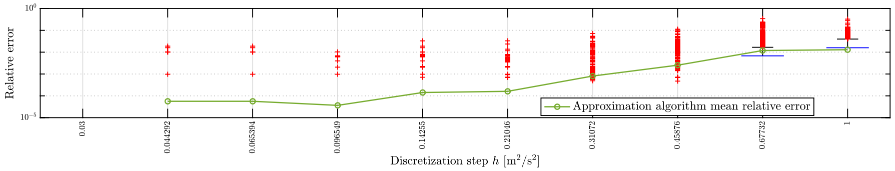


Fig. 17 Approximation algorithm and speed planning algorithm absolute error for different values of *h* on the second scenario

Figure 17 shows the box-and-whisker plot of the relative error on the travel time of the trajectories obtained coupling the approximation algorithm of Sect. 5 with the speed planning one. Again, we set a tolerance on relative errors of 10^{-4}. In this case the mean relative errors are on average two orders of magnitude smaller than those presented in Fig. 15.

For this scenario, if we fix $h = 0.21046\,\mathrm{m}^2\,\mathrm{s}^{-2}$, we get a mean computational time that is 99.24 times faster than that of the adaptive A* algorithm for *k*-BASP, while obtaining the exact solution up to a tolerance of 10^{-4} on the relative error in 97.6% of the cases.

Observe that the construction of the extended graphs can be a time-consuming operation. One could alternatively run a dynamic programming approach by generating arcs only when (and if) needed. However, the construction has to be performed only once and the extended graphs can be stored for future use. The memory occupancy of the extended graphs of the scenarios we considered varies from 36 KB for G'_h with $h = 0.5$ m^2 s^{-2} to 258 MB for G'_h with $h = 0.005$ m^2 s^{-2} for the first scenario, and from 78 KB for G'_h with $h = 1\,\mathrm{m}^2\,\mathrm{s}^{-2}$ to 56.1 MB for G'_h with $h = 0.03\,\mathrm{m}^2\,\mathrm{s}^{-2}$ for the second scenario.

7 Conclusions

Motivated by an industrial application, in this paper we addressed a variant of the Shortest Path Problem (SPP). The variant is called BASP (Bounded Acceleration SP) since speed and acceleration constraints are imposed over the arcs. Differently from SPP, where the traveling time of an arc is constant, in BASP the traveling time depends on the initial and final speed along the arc and, thus, due to the speed and acceleration constraints, it also depends on the arcs preceding and following it along a path. We proved that BASP is NP-hard but also that, under the assumption of integer data, it admits a pseudo-polynomial time algorithm. We also proposed an approximation algorithm based on the solution of an SPP problem over an extended graph. The extended graph is defined by discretizing the admissible speeds at the nodes of the graph. Finally, we performed different computational experiments on two real-life industrial scenarios in order to evaluate the performance of the approximation algorithm.

Appendix

A Proof of Proposition 4.1

Each path P from node 0 to node $n+2$ has the following structure

$$0 \to i_1 \to i_2 \to \cdots \to i_r \to n+1 \to n+2,$$

with $i_1 < i_2 < \cdots < i_r$. Let us denote by $N_P = \{i_1, i_2, \dots, i_r\}$ the set of intermediate nodes in P. The length of path P is $W^2 + \ell(P)$, where $\ell(P) = \sum_{i \in N_P} \beta_i$ is the length of the path up to node $n+1$.

Before proceeding with the proof we give the intuition behind it. We will show that for each path with length $\ell(P) > \frac{W}{2}$ up to node $n+1$, the traveling time from node 0 to node $n+2$ is larger than the traveling time of a path with length $\ell(P) = \frac{W}{2}$ up to node $n+1$ (if any). This will simply follow from the fact that the former path is longer and the speed along the final arc $(n+1, n+2)$ is the same in both cases. Moreover, we will show that also for a path with length $\ell(P) < \frac{W}{2}$ up to node $n+1$, the traveling time from node 0 to node $n+2$ is larger than the traveling time of a path with length $\ell(P) = \frac{W}{2}$ up to node $n+1$ (again, if any). In this case this will be proved by observing that the speed reached along this path at node $n+1$ will be lower than $\sqrt{W}$ (the speed reached by any path with length $\ell(P) \geq \frac{W}{2}$). Since it is not possible to accelerate along arc $(n+1, n+2)$, the traveling time along this arc will be higher with respect to paths with length $\ell(P) \geq \frac{W}{2}$ up to node $n+1$. In particular, we will show that the increase of the time needed to travel along arc $(n+1, n+2)$ will be larger than the time saved along the sub-path from node 0 to node $n+1$ with respect to a path for which $\ell(P) = \frac{W}{2}$. In conclusion, we will show that the Partition problem has a `yes` answer if and only if the optimal value of the BASP instance is the traveling time of a path with length $\ell(P) = \frac{W}{2}$ up to node $n+1$. In what follows we prove the result in a formal way.

Let us first assume that $\ell(P) < \frac{W}{2}$. In this case, according to the discussion in Sect. 3, the maximum speed which can be reached at node $n+1$ is $v_u = a_{\max} t_P$, where $a_{\max} = 1$ and t_P fulfills $\ell(P) = \frac{1}{2} a_{\max} t^2$, that is, $t_P = \sqrt{2\ell(P)}$. Thus, $v_u = \sqrt{2\ell(P)} < \sqrt{W}$. Along the final arc of the path $(n+1, n+2)$ the maximum acceleration is null, while the maximum deceleration is $-\frac{1}{2W}$. Then, the optimal speed profile along this arc is obtained by keeping the speed $v_u = \sqrt{2\ell(P)}$ for a portion of the arc with length

$W^2 - 2\ell(P)W$, while in the last portion, with length $2\ell(P)W$, the speed is decreased with the maximum possible deceleration $-\frac{1}{2W}$. Now we denote by $t^1_{L_1}(\ell(P))$ the time to run along the first portion of the arc, while we denote by $t_{L_2}(\ell(P))$ the time to run along the second part of the arc. We have that

$$t_{L_1}(\ell(P)) = \frac{W^2 - 2\ell(P)W}{\sqrt{2\ell(P)}},$$

while $t_{L_2}(\ell(P))$ fulfills the following condition

$$\sqrt{2\ell(P)}W = \sqrt{2\ell(P)}t_{L_2}(\ell(P)) - \frac{1}{4W}t_{L_2}(\ell(P))^2,$$

that is,

$$t_{L_2}(\ell(P)) = 2W\sqrt{2\ell(P)}.$$

Thus, the overall time to traverse such paths is

$$T_1(\ell(P)) = \sqrt{2\ell(P)} + \frac{W^2 - 2\ell(P)W}{\sqrt{2\ell(P)}} + t_{L_2}(\ell(P)) = \sqrt{2\ell(P)} + \frac{W^2}{\sqrt{2\ell(P)}} + W\sqrt{2\ell(P)}.$$

Now, let us consider paths P such that $\ell(P) \geq \frac{W}{2}$. A lower bound for the time needed to run along the path up to node $n+1$ is given again by the solution of the following simple equation $\ell(P) = \frac{1}{2}a_{\max}t^2$, which is $t_P = \sqrt{2\ell(P)}$. Note that this is a lower bound since with the maximum acceleration we would reach the speed $v_u = a_{\max}t_P = \sqrt{2\ell(P)} \geq \sqrt{W}$, so that we stop accelerating as soon as we reach the maximum speed $\sqrt{W}$. Since $\ell(P) \geq \frac{W}{2}$, we have that the lower bound can be further bounded from below by $\sqrt{W}$. Finally, we observe that such lower bound can be attained if and only if $\ell(P) = \frac{W}{2}$, that is, if and only if the Partition problem admits a solution. Now, over the last arc $(n+1, n+2)$ we can decrease the speed to 0 at node $n+2$ by keeping the maximum deceleration $-\frac{1}{2W}$ over the whole arc. Then, the time needed to traverse this arc, denoted by t_{L_3}, fulfills

$$W^2 = \sqrt{W}t_{L_3} - \frac{1}{4W}(t_{L_3})^2,$$

so that $t_{L_3} = 2W^{3/2}$. Then, a lower bound for the time needed to traverse such paths is $T_2 = \sqrt{W} + 2W^{3/2}$ and, as already pointed out, such lower bound is attained if and only if $\ell(P) = \frac{W}{2}$. Figure 19 illustrates the optimal speed profiles for three distinct paths P_1, P_2 and P_3, fulfilling $\ell(P_1) < \frac{W}{2}$, $\ell(P_2) = \frac{W}{2}$, and $\ell(P_3) > \frac{W}{2}$, respectively. The three profiles are depicted, respectively with a dashed, a continuous and a dotted line (up to distance $\ell(P_1)$ they are overlapping). We notice that for the shortest path P_1 we are unable to reach the maximum squared speed W, so that along arc $(n+1, n+2)$ we first increase the speed as much as possible with acceleration

$a^{\max}_{n+1,n+2}$ (in fact, we keep the speed constant since $a^{\max}_{n+1,n+2} = 0$), and then we decrease the speed as fast as possible with deceleration $a^{\min}_{n+1,n+2}$. For the intermediate path P_2 we reach the maximum squared speed exactly at the end of the path and then, along arc $(n+1, n+2)$ we decrease the speed as fast as possible with deceleration $a^{\min}_{n+1,n+2}$. Finally, for the longest path P_3 we reach the maximum squared speed before the end of the path, and we keep the speed constant in the last part of the path, while along arc $(n+1, n+2)$ we decrease the speed as fast as possible with deceleration $a^{\min}_{n+1,n+2}$.

Now, if we are able to prove that $T_2 < T_1(\ell(P))$ when $\ell(P) \leq \frac{W}{2} - 1$, then we are done. We have that

$$T_2 - T_1(\ell(P)) = \sqrt{W} + 2W^{3/2} - W\sqrt{2\ell(P)} - \sqrt{2\ell(P)} - \frac{W^2}{\sqrt{2\ell(P)}}.$$

By the change of variable $x = \sqrt{2\ell(P)}$, this can be rewritten as

$$\frac{-(W+1)x^2 + (\sqrt{W} + 2W^{3/2})x - W^2}{x},$$

which can be easily seen to be negative for all $x \leq \frac{W^{3/2}}{W+1}$. Now, recalling the definition of x, this means that $T_2 - T_1(\ell(P))$ is negative if

$$\ell(P) < \frac{W^3}{2(W+1)^2}.$$

Then, the result follows by observing that

$$\frac{W^3}{2(W+1)^2} > \frac{W}{2} - 1.$$

B Proof of Lemma 5.2

For $i \in \{1, \dots, |P|\}$, set $a_i^+ = 2a^{\max}_{P(i),P(i+1)}\ell_{P(i),P(i+1)}$, $a_i^- = 2a^{\min}_{P(i),P(i+1)}\ell_{P(i),P(i+1)}$. For $i \in \{2, \dots, |P|\}$, set $w_i^+ = \min\{w^{\max}_{P(i-1),P(i)}, w^{\max}_{P(i),P(i+1)}\}$. It is convenient to rewrite Problem (5) as:

$$\begin{aligned} T(P) = \min_{\mathbf{w}\in\mathbb{R}^{|P|+1}} & \sum_{i=1}^{|P|} c_{P(i)\,P(i+1)}(w_i, w_{i+1}) & \\ & 0 \leq w_i \leq w_i^+, & i \in \{2, \dots, |P|\}, \\ & w_{i+1} \leq w_i + a_i^+, & i \in \{1, \dots, |P|\}, \\ & w_i \leq w_{i+1} - a_i^-, & i \in \{1, \dots, |P|\}, \\ & w_1 = 0,\ w_{|P|+1} = 0\,. & \end{aligned} \tag{13}$$

Problem (13) belongs to the class of Problem 5 in [3] (nondecreasing right-hand side of the constraints and strictly monotonically decreasing objective function) . As reported there, the solution of Problem (5) can be found as follows. Define vectors $\mathbf{F}, \mathbf{B}, \mathbf{W} \in \mathbb{R}^{|P|+1}$ such that:

$$\begin{array}{ll}
F_1 = 0 & \\
F_i = \min\{F_{i-1} + a^+_{i-1}, w^+_i\}, & i \in \{2, \dots, |P|+1\}, \\
[6pt]B_{|P|+1} = 0 & \\
[6pt]B_i = \min\{B_{i+1} - a^-_i, w^+_i\}, \; i \in \{1, \dots, |P|\}, & \\
[6pt]\mathbf{W} = \min\{\mathbf{F}, \mathbf{B}\}. &
\end{array}$$

Note that the components of vector $\mathbf{W}$ are equivalent to the values of function $W(s)$ defined in Sect. 3 evaluated at nodes of the path, while along each arc of the path function $W(s)$ is defined according to (1). As a consequence of Theorem 2 of [3] the solution of Problem (13) is $\mathbf{w}^*(P) = \mathbf{W}$. On the other hand, Problem (8) can be rewritten in the following form:

$$\begin{array}{lll}
T^h(P) = \min\limits_{\mathbf{w} \in \mathbb{R}^{|P|+1}} \sum\limits_{i=1}^{|P|} c_{P(i)P(i+1)}(w_i, w_{i+1}) & & \\
\qquad 0 \leq w_i \leq \langle w^+_i \rangle, & i \in \{2, \dots, |P|\}, & \\
\qquad w_{i+1} \leq \langle w_i + a^+_i \rangle, & i \in \{1, \dots, |P|\}, & \\
\qquad w_i \leq \langle w_{i+1} - a^-_i \rangle, & i \in \{1, \dots, |P|\}, & \\
\qquad w_1 = 0, \; w_{|P|+1} = 0\,. & &
\end{array} \tag{14}$$

Note that, strictly speaking, this is a relaxation of Problem (8), since we have not included the last constraint, namely $\langle w_i \rangle = w_i, i \in \{1, \dots, |P|+1\}$. In fact, as shown in [3], at least one constraint is active at each component $w^h_i(P), i \in \{1, \dots, |P|+1\}$, of the optimal solution of (14). Since $0 = \langle 0 \rangle$ and, for all $x \in \mathbb{R}$, $\langle \langle x \rangle \rangle = \langle x \rangle$, necessarily $\langle w^h_i(P) \rangle = w^h_i(P)$, $i \in \{1, \dots, |P|+1\}$ (which means that the optimal solution of (14) fulfills the last constraint in (8)). Also Problem (14) belongs to the class of Problem 5 in [3]. Hence, its solution can be computed with the same procedure used for Problem (13). Namely, define vectors $\mathbf{F}^h, \mathbf{B}^h, \mathbf{W}^h \in \mathbb{R}^{|P|+1}$ such that:

$$\begin{array}{l}
F^h_1 = 0 \\
F^h_i = \langle \min\{F^h_{i-1} + a^+_{i-1}, w^+_i\} \rangle, \; i \in \{2, \dots, |P|+1\}, \\
B^h_{|P|+1} = 0 \\
B^h_i = \langle \min\{B^h_{i+1} - a^-_{i+1}, w^+_i\} \rangle, \; i \in \{1, \dots, |P|\}, \\
\mathbf{W}^h = \min\{\mathbf{F}^h, \mathbf{B}^h\}.
\end{array}$$

Again, by Theorem 2 in [3], $\mathbf{w}^h(P) = \mathbf{W}^h$. Note that, by their definitions, $\mathbf{B}^h \leq \mathbf{B}$ and $\mathbf{F}^h \leq \mathbf{F}$.

Now we are ready to prove Lemma 5.2. We first prove that for all $i \in \{1, \dots, |P|+1\}$:

$$F_i - F^h_i \leq h(i-1).$$

Set $\delta_i = F_i - F_i^h$. By the definition of F_i^h we may have that $F_i^h = \langle F_{i-1}^h + a_{i-1}^+ \rangle$ or $F_i^h = \langle w_i^+ \rangle$. In the first case, $\delta_i \le F_{i-1} - F_{i-1}^h + a_{i-1}^+ - \langle a_{i-1}^+ \rangle \le \delta_{i-1} + h$. In the second case, $\delta_i \le w_i^+ - \langle w_i^+ \rangle \le h$. Hence, $\boldsymbol{\delta}$ satisfies

$$\begin{aligned}&\delta_1 = 0\\ &\delta_{i+1} \le \delta_i + h, \quad i \in \{1, \dots, |P|\},\end{aligned}$$

and it follows that $\delta_i \le h(i-1)$. Proceeding in the same way as above, setting $\eta_i = B_i - B_i^h$, $\boldsymbol{\eta}$ satisfies

$$\begin{aligned}&\eta_{|P|+1} = 0\\ &\eta_i \le \eta_{i+1} + h, \quad i \in \{1, \dots, |P|\},\end{aligned}$$

and, consequently:

$$B_i - B_i^h \le h(|P| + 1 - i).$$

Finally, we observe that

$$w^*(P)_i - w^h(P)_i = \min\{F_i, B_i\} - \min\{F_i^h, B_i^h\} \le \max\{F_i - F_i^h, B_i - B_i^h\} \le h|P|,$$

as we wanted to prove.

C Proof of Lemma 5.4

Given an arc (i, j) and squared speeds $w, z, \hat{w}, \hat{z} \in \mathbb{R}$, with $0 \le w, z, \hat{w}, \hat{z} \le w_{ij}^{\max}$, $w \le \hat{w}$ and $z \le \hat{z}$, for an interval $[a, b] \subseteq [0, \ell_{ij}]$, define

$$e_{[a,b]} = \left| \int_a^b \left(\frac{1}{\sqrt{w_{ij}(s; w, z)}} - \frac{1}{\sqrt{w_{ij}(s; \hat{w}, \hat{z})}} \right) ds \right|,$$

where w_{ij} is defined in (1). Our goal is to find an upper bound for $\left| c_{ij}(w, z) - c_{ij}(\hat{w}, \hat{z}) \right| = e_{[0,\ell_{ij}]}$. We consider only the case $\hat{w} \le \hat{z}$ (the converse case in which $\hat{w} \ge \hat{z}$ is analogous).

We find a bound on $\left| c_{ij}(w, z) - c_{ij}(\hat{w}, \hat{z}) \right|$ by splitting the integration interval $[0, \ell_{ij}]$ into three parts: the initial part $[0, \bar{s}]$, with $\bar{s} \in [0, \ell_{ij}]$, in which both the speed profiles starting at $\hat{w}$ and w are growing, the second part $[\bar{s}, \tilde{s}]$, with $\tilde{s} \in [\bar{s}, \ell_{ij}]$ in which the speed profile starting at $\hat{w}$ keeps growing whilst the one starting at w starts decreasing, and the final part $[\tilde{s}, \ell_{ij}]$, in which both speed profiles are decreasing (see Fig. 18). In case the first profile never decreases, we set $\tilde{s} = \ell_{ij}$, and in case the second profile never decreases, we set $\tilde{s} = \bar{s} = \ell_{ij}$. In this way, $\left| c_{ij}(w, z) - c_{ij}(\hat{w}, \hat{z}) \right| = e_{[0,\bar{s}]} + e_{[\bar{s},\tilde{s}]} + e_{[\tilde{s},\ell_{ij}]}$. We will need the following technical remark.

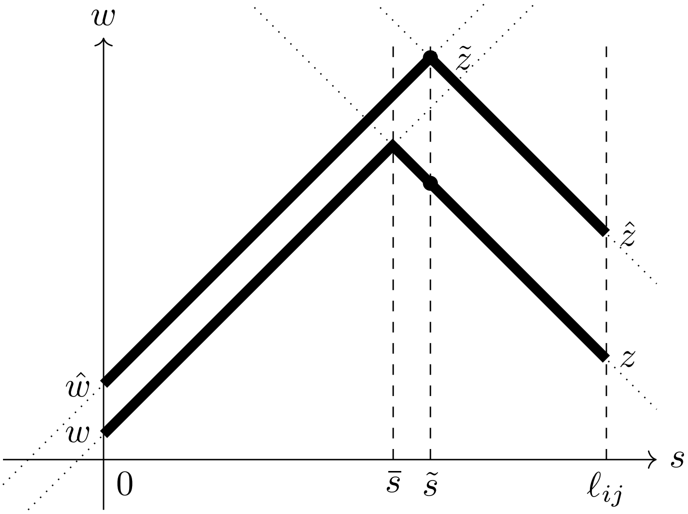


Fig. 18 Comparison on a generic arc between the optimal speed profile starting from $\hat{w}$ and ending at $\hat{z}$ and the one starting from w and ending at z

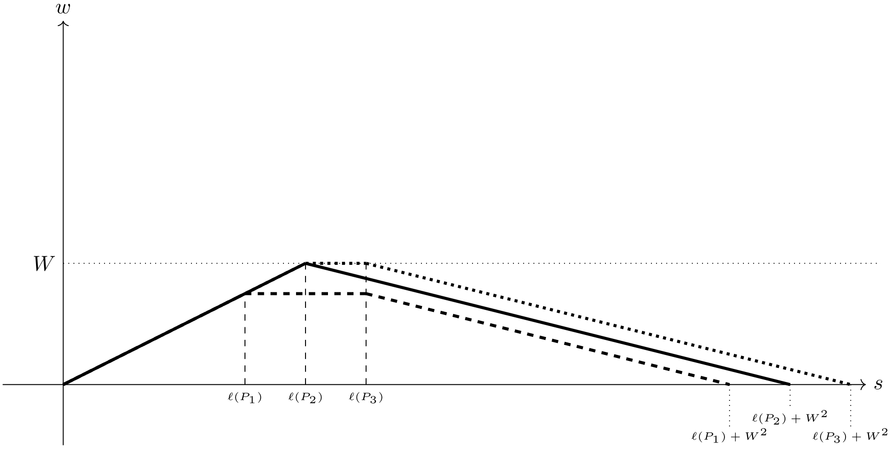


Fig. 19 Maximum (squared) speed profiles along three paths P_1, P_2, P_3 fulfilling $\ell(P_1) < \frac{W}{2}$, $\ell(P_2) = \frac{W}{2}$, and $\ell(P_3) > \frac{W}{2}$

Remark C.1 Given $c > 0$ and $x \in [0, c]$, it holds that

$$-\frac{x}{\sqrt{c}} + \sqrt{c} \le \sqrt{c - x} \le -\frac{x}{2\sqrt{c}} + \sqrt{c}.$$

Next lemma establishes a bound from above for the error along the first part of the arc.

Lemma C.1 *Given arc* (i, j) *and* $0 \leq w, \hat{w}, z, \hat{z}$, *with* $w \leq \hat{w}$ *and* $z \leq \hat{z}$ *the following bound holds*:

$$\begin{cases} e_{[0,\bar{s}]} = 0 & \text{if } \hat{w} = 0, \\ e_{[0,\bar{s}]} \leq \frac{|w-\hat{w}|}{a_{ij}^{\max}\sqrt{\hat{w}}} & \text{otherwise .} \end{cases}$$

Proof If $\hat{w} = 0$, then also $w = 0$, hence

$$e_{[0,\bar{s}]} = \int_0^{\bar{s}} \left(\frac{1}{\sqrt{w + 2a_{ij}^{\max} s}} - \frac{1}{\sqrt{\hat{w} + 2a_{ij}^{\max} s}} \right) ds = \int_0^{\bar{s}} \left(\frac{1}{\sqrt{2a_{ij}^{\max} s}} - \frac{1}{\sqrt{2a_{ij}^{\max} s}} \right) ds = 0.$$

Otherwise, if $\hat{w} > 0$, we can bound the error over $[0, \bar{s}]$ as follows

$$\begin{aligned} e_{[0,\bar{s}]} &= \int_0^{\bar{s}} \left(\frac{1}{\sqrt{w + 2a_{ij}^{\max} s}} - \frac{1}{\sqrt{\hat{w} + 2a_{ij}^{\max} s}} \right) ds = \left[\frac{\sqrt{w + 2a_{ij}^{\max} s}}{a_{ij}^{\max}} - \frac{\sqrt{\hat{w} + 2a_{ij}^{\max} s}}{a_{ij}^{\max}} \right]_0^{\bar{s}} \\ &= \frac{1}{a_{ij}^{\max}} \left(\sqrt{w + 2a_{ij}^{\max} \bar{s}} - \sqrt{w} - \sqrt{\hat{w} + 2a_{ij}^{\max} \bar{s}} - \sqrt{\hat{w}} \right) \\ &\leq \frac{1}{a_{ij}^{\max}} \left(- \frac{|w - \hat{w}|}{2\sqrt{\hat{w} + 2a_{ij}^{\max} \bar{s}}} + \sqrt{\hat{w} + 2a_{ij}^{\max} \bar{s}} + \frac{|w - \hat{w}|}{\sqrt{\hat{w}}} - \sqrt{\hat{w}} - \sqrt{\hat{w} + 2a_{ij}^{\max} \bar{s}} + \sqrt{\hat{w}} \right) \\ &= \left(\frac{1}{\sqrt{\hat{w}}} - \frac{1}{2\sqrt{\hat{w} + 2a_{ij}^{\max} \bar{s}}} \right) \frac{|w - \hat{w}|}{a_{ij}^{\max}} \leq \frac{|w - \hat{w}|}{a_{ij}^{\max}\sqrt{\hat{w}}}, \end{aligned}$$

where, in the first inequality, we used the fact that, for $|w - \hat{w}| \leq \hat{w} + 2a_{ij}^{\max}\bar{s}$, in view of Remark C.1 it holds that:

$$\sqrt{w + 2a_{ij}^{\max}\bar{s}} = \sqrt{\hat{w} - |w - \hat{w}| + 2a_{ij}^{\max}\bar{s}} \leq -\frac{|w - \hat{w}|}{2\sqrt{\hat{w} + 2a_{ij}^{\max}\bar{s}}} + \sqrt{\hat{w} + 2a_{ij}^{\max}\bar{s}},$$

and that, for $|w - \hat{w}| \leq \hat{w}$,

$$\sqrt{w} = \sqrt{\hat{w} - |w - \hat{w}|} \geq -\frac{|w - \hat{w}|}{\sqrt{\hat{w}}} + \sqrt{\hat{w}}.$$

□

The error over the second part $[\bar{s}, \tilde{s}]$ of arc (i, j) can be bounded from above as follows.

Lemma C.2 *Given arc* (i, j) *and* $0 \leq w, \hat{w}, z, \hat{z}$, *with* $w \leq \hat{w}$, $z \leq \hat{z}$, *the following bound holds*:

$$e_{[\bar{s},\tilde{s}]} \leq \max\{|w-\hat{w}|, |z-\hat{z}|\}\frac{2}{|a_{ij}^{\min}|\sqrt{\tilde{z}}},$$

where $\tilde{z} = w_{ij}(\tilde{s};\hat{w},\hat{z})$ is the squared speed at $\tilde{s}$ for the profile with boundary speeds $\hat{w},\hat{z}$.

Proof We can bound from above the error over $[\bar{s},\tilde{s}]$ as follows

$$\begin{aligned}
e_{[\bar{s},\tilde{s}]} &= \int_{\bar{s}}^{\tilde{s}} \left(\frac{1}{\sqrt{\tilde{z} - |z-\hat{z}| + 2a_{ij}^{\min}(s-\tilde{s})}} - \frac{1}{\sqrt{\hat{w} + 2a_{ij}^{\max} s}} \right) ds \\
&\leq \int_{\bar{s}}^{\tilde{s}} \frac{ds}{\sqrt{\tilde{z} - |z-\hat{z}| + 2a_{ij}^{\min}(s-\tilde{s})}} \\
&= \left[\frac{\sqrt{\tilde{z} - |z-\hat{z}| + 2a_{ij}^{\min}(s-\tilde{s})}}{a_{ij}^{\min}} \right]_{\bar{s}}^{\tilde{s}} \\
&= \frac{1}{|a_{ij}^{\min}|}\left(\sqrt{\tilde{z} - |z-\hat{z}| + 2a_{ij}^{\min}(\bar{s}-\tilde{s})} - \sqrt{\tilde{z} - |z-\hat{z}|} \right) \\
&\leq \frac{1}{|a_{ij}^{\min}|}\left(-\frac{|z-\hat{z}|}{2\sqrt{\tilde{z} + 2a_{ij}^{\min}(\bar{s}-\tilde{s})}} + \sqrt{\tilde{z} + 2a_{ij}^{\min}(\bar{s}-\tilde{s})} + \frac{|z-\hat{z}|}{\sqrt{\tilde{z}}} - \sqrt{\tilde{z}} \right) \\
&\leq \frac{1}{|a_{ij}^{\min}|}\left(\sqrt{\tilde{z} + 2a_{ij}^{\min}(\bar{s}-\tilde{s})} + \frac{|z-\hat{z}|}{\sqrt{\tilde{z}}} - \sqrt{\tilde{z}} \right) \\
&\leq \frac{1}{|a_{ij}^{\min}|}\left(\sqrt{\tilde{z} + \max\{|w-\hat{w}|, |z-\hat{z}|\}} + \frac{|z-\hat{z}|}{\sqrt{\tilde{z}}} - \sqrt{\tilde{z}} \right) \\
&\leq \frac{1}{|a_{ij}^{\min}|}\left(\frac{\max\{|w-\hat{w}|, |z-\hat{z}|\}}{2\sqrt{\tilde{z}}} + \sqrt{\tilde{z}} + \frac{|z-\hat{z}|}{\sqrt{\tilde{z}}} - \sqrt{\tilde{z}} \right) \\
&\leq \max\{|w-\hat{w}|, |z-\hat{z}|\}\frac{2}{|a_{ij}^{\min}|\sqrt{\tilde{z}}},
\end{aligned}$$

where, in the second inequality, we used the fact that, for $|z-\hat{z}| \leq \tilde{z} + 2a_{ij}^{\min}(\bar{s}-\tilde{s})$, in view of Remark C.1, it holds that

$$\sqrt{\tilde{z} - |z-\hat{z}| + 2a_{ij}^{\min}(\bar{s}-\tilde{s})} \leq -\frac{|z-\hat{z}|}{2\sqrt{\tilde{z} + 2a_{ij}^{\min}(\bar{s}-\tilde{s})}} + \sqrt{\tilde{z} + 2a_{ij}^{\min}(\bar{s}-\tilde{s})},$$

and that, for $|z-\hat{z}| \leq \tilde{z}$,

$$\sqrt{\tilde{z}-|z-\hat{z}|} \geq -\frac{|z-\hat{z}|}{\sqrt{\tilde{z}}}+\sqrt{\tilde{z}},$$

whilst, in the fourth inequality, we used the fact that $2a_{ij}^{\min}(\bar{s}-\tilde{s}) \leq \max\{|w-\hat{w}|,|z-\hat{z}|\}$. Indeed, the following equations hold (see also Fig. 18)

$$\begin{aligned} \hat{z}+2a_{ij}^{\min}(\tilde{s}-\ell_{ij}) &= \hat{w}+2a_{ij}^{\max}\tilde{s} \\ [6pt]z+2a_{ij}^{\min}(\bar{s}-\ell_{ij}) &= w+2a_{ij}^{\max}\bar{s}. \end{aligned}$$

Taking the difference between the second and the first equation, we end up with

$$\begin{aligned} 2a_{ij}^{\min}(\bar{s}-\tilde{s}) &= |z-\hat{z}|-|w-\hat{w}|+2a_{ij}^{\max}(\bar{s}-\tilde{s}) \\ &\leq |z-\hat{z}|-|w-\hat{w}| \leq \max\{|w-\hat{w}|,|z-\hat{z}|\}, \end{aligned}$$

where the first inequality comes from $a_{ij}^{\max} > 0$ and $\bar{s} \leq \tilde{s}$. □

Finally, we establish a bound for the error along the third part of the arc.

Lemma C.3 *Given arc* (i, j) *and* $0 \leq w, \hat{w}, z, \hat{z}$, *with* $w \leq \hat{w}$, $z \leq \hat{z}$, *the following bound holds*:

$$\begin{cases} e_{[\tilde{s},\ell_{ij}]} = 0 & \text{if } \hat{z}=0, \\ e_{[\tilde{s},\ell_{ij}]} \leq \frac{|z-\hat{z}|}{|a_{ij}^{\min}|\sqrt{\hat{z}}} & \text{otherwise.} \end{cases}$$

Proof The proof of Lemma C.3 is analogous to that of Lemma C.1. □

Now, we can prove Lemma 5.4, that is, we can provide an estimate on the absolute error over the entire arc (i, j).

Proof By Lemmas C.1–C.3, we have the following cases. If $\hat{w}=\hat{z}=0$, then

$$c_{ij}(w,z)-c_{ij}(\hat{w},\hat{z})=0$$

(note that in this case $\bar{s}=\tilde{s}$, so that $e_{[\bar{s},\tilde{s}]}=0$). If $\hat{w}=0$, then:

$$c_{ij}(w,z)-c_{ij}(\hat{w},\hat{z})=e_{[\bar{s},\tilde{s}]}+e_{[\tilde{s},\ell_{ij}]} \leq |z-\hat{z}|\frac{2}{|a_{ij}^{\min}|\sqrt{\tilde{z}}}+\frac{|z-\hat{z}|}{|a_{ij}^{\min}|\sqrt{\hat{z}}} \leq |z-\hat{z}|\frac{3}{|a_{ij}^{\min}|\sqrt{\hat{z}}},$$

where the last inequality follows from $\tilde{z} \geq \hat{z}$. Similarly, for $\hat{z}=0$:

$$\begin{aligned} c_{ij}(w,z)-c_{ij}(\hat{w},\hat{z}) =& e_{[0,\bar{s}]}+e_{[\bar{s},\tilde{s}]} \\ &\leq \frac{|w-\hat{w}|}{a_{ij}^{\max}\sqrt{\hat{w}}}+|w-\hat{w}|\frac{2}{|a_{ij}^{\min}|\sqrt{\tilde{z}}} \leq |w-\hat{w}|\frac{3}{\min\{a_{ij}^{\max},-a_{ij}^{\min}\}\sqrt{\hat{w}}}, \end{aligned}$$

where the last inequality follows from $\tilde{z} \geq \hat{w}$. Finally, if $\hat{w}, \hat{z} > 0$:

$$\begin{aligned} c_{ij}(w,z) - c_{ij}(\hat{w},\hat{z}) =& e_{[0,\bar{s}]} + e_{[\bar{s},\tilde{s}]} + e_{[\tilde{s},\ell_{ij}]} \\ \leq& \frac{|w-\hat{w}|}{a_{ij}^{\max}\sqrt{\hat{w}}} + \max\{|w-\hat{w}|,|z-\hat{z}|\}\frac{2}{|a_{ij}^{\min}|\sqrt{\tilde{z}}} + \frac{|z-\hat{z}|}{|a_{ij}^{\min}|\sqrt{\hat{z}}} \\ \leq& \frac{|w-\hat{w}| + 2\max\{|w-\hat{w}|,|z-\hat{z}|\} + |z-\hat{z}|}{\min\{a_{ij}^{\max}, -a_{ij}^{\min}\}\sqrt{\min\{\hat{w},\hat{z}\}}} \\ =& \frac{4\max\{|w-\hat{w}|,|z-\hat{z}|\}}{\min\{a_{ij}^{\max}, -a_{ij}^{\min}\}\sqrt{\min\{\hat{w},\hat{z}\}}}. \end{aligned}$$

Now the results immediately follows from the fact that, by the definitions of $a_{\min}$ and $\bar{w}$, it holds that $a_{ij}^{\max}, |a_{ij}^{\min}| \geq a_{\min}$, $\hat{w} \geq \bar{w}$ if $\hat{w} > 0$, and $\hat{z} \geq \bar{w}$ if $\hat{z} > 0$. □

Funding Open access funding provided by Università degli Studi di Parma within the CRUI-CARE Agreement.

Data availability The data and code that support the findings of this study are available at: OSF. May 23. osf.io/6jupx S. Ardizzoni, L. Consolini, M. Laurini, and M. Locatelli. 2022. “Bounded Acceleration Shortest Path Problem”.

References

1. Ardizzoni, S., Consolini, L., Laurini, M., Locatelli, M.: Solution algorithms for the bounded acceleration shortest path problem. IEEE Trans. Autom. Control (2022) (to appear). https://doi.org/10.1109/TAC.2022.3172169
2. Chen, B.Y., Chen, X.-W., Chen, H.-P., Lam, W.H.K.: Efficient algorithm for finding *k* shortest paths based on re-optimization technique. Transpt. Res. Part E: Log. Transpt. Rev. **133**, 101819 (2020)
3. Consolini, L., Locatelli, M., Minari, A., Piazzi, A.: An optimal complexity algorithm for minimum-time velocity planning. Syst. Control Lett. **103**, 50–57 (2017)
4. Consolini, L., Laurini, M., Locatelli, M., Minari, A.: A solution of the minimum-time speed planning problem based on lattice theory. J. Frankl. Inst. **357**(12), 7617–7637 (2020)
5. Cowlagi, R.V., Tsiotras, P.: Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning. In: 2009 American Control Conference, pp. 414–419 (2009)
6. Dean, B.C.: Shortest paths in FIFO time-dependent networks: theory and algorithms. In: Technical Report Massachusetts Institute of Technology (2004)

7. Desaulniers, G., Villeneuve, D.: The shortest path problem with forbidden paths. Eur. J. Oper. Res. **165**, 97–107 (2005)
8. Desrochers, M., Soumis, F.: A reoptimization algorithm for the shortest path problem with time windows. Eur. J. Oper. Res. **35**, 242–254 (1988)
9. Dreyfus, S.: An Appraisal of Some Shortest-Path Algorithms. Oper. Res. **17**, 395–412 (1969)
10. Dror, M.: Note on the complexity of the shortest path models for column generation in VRPTW. Oper. Res. **42**, 977–978 (1994)
11. Edmonds, J., Fulkerson, D.R.: Bottleneck extrema. J. Comb. Theory **8**(3), 299–306 (1970)
12. Festa, P.: Constrained shortest path problems: state-of-the-art and recent advances. In: 17th International Conference on Transparent Optical Networks (ICTON), pp. 1–17 (2015)
13. Garfinkel, R., Fernandez, E., Lowe, T.J.: The *k*-centrum shortest path problem. TOP **14**(2), 279–292 (2006)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
15. Handler, G.Y., Zang, I.: A dual algorithm for the constrained shortest path problem. Networks **10**(4), 293–309 (1980)
16. He, E., Boland, N., Nehmauser, G., Savelsbergh, M.: Time-dependent shortest path problems with penalties and limits on waiting. INFORMS J. Comput. **33**(3), 997–1014 (2021)
17. Ioachim, I., Gélinas, S., Soumis, F., Desrosiers, J.: A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. Networks **31**, 193–204 (1998)
18. Omer, J., Poss, M.: Time-dependent shortest paths with discounted waits. Networks **4**(3), 287–301 (2019)
19. Orda, A., Rom, R.: Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. J. ACM **37**, 607–625 (1990)
20. Pollack, M.: The maximum capacity through a network. Oper. Res. **8**(5), 733–736 (1960)
21. Turner, L.: Variants of shortest path problems. Algorithm. Oper. Res. **6**(2), 91–104 (2012)
22. Stern, R., Sturtevant, N.R., Felner, A., Keonig, S., Ma, H., Walker, T.T., Li, J., Atzmon, D., Cohen, L., Kumar, T.S., Boyarski, E., Bartak, R.: Multi-agent pathfinding: definitions, variants, and benchmarks. In: Symposium on Combinatorial Search (SoCS) (2019)