International Journal of Approximate Reasoning 88 (2017) 567–586

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Influence diagrams for speed profile optimization ☆

Václav Kratochvíl, Jiří Vomlel *

Institute of Information Theory and Automation, Czech Academy of Sciences, Czechia

ARTICLE INFO

Article history:
Received 31 March 2016
Received in revised form 11 October 2016
Accepted 25 November 2016
Available online 2 December 2016

Keywords:
Influence diagrams
Optimal control
Vehicle control

ABSTRACT

Influence diagrams have been applied to diverse decision problems. In this paper, we describe their application to the speed profile optimization problem – a problem traditionally solved by the methods of optimal control theory.

Influence diagrams appeared to be well-suited to these types of problems. It is mainly due to their ability to perform computations efficiently if the utility function is additively decomposed along the vehicle path, which is the case for utility functions based on, e.g., the total driving time or the total fuel consumption. Also, driving constraints can be efficiently included in the influence diagram. If the vehicle speed deviates from the optimal speed profile during the real drive, a new optimal speed profile can be quickly computed in the compiled influence diagram.

The theory of influence diagrams has not yet been sufficiently developed for continuous variables and nonlinear utility functions. We cope with this issue by discretization and by stochastic approximations of deterministic problems.

We performed experiments on a real problem – the speed control of a Formula 1 race car. Influence diagrams can provide a good solution of the problem very quickly. This solution can be used as an initial solution for the methods of the optimal control theory and improves the convergence of these methods.

1. Introduction

Optimization of a vehicle speed profile is a well-known problem studied in the literature. Some authors minimize the energy consumption [16,3,18,4,14,17] while others aim at minimizing the total time [24]. Traditionally, this problem is solved by methods of the optimal control theory [10].

In this paper we describe an application of influence diagrams [7,19,21,9] to the problem of the optimization of a vehicle speed profile, which specifies the recommended vehicle speed at each point on the path. This paper extends our results previously published in two conference papers [11] and [25].

There are two key properties that allow efficient computations with influence diagrams. The first one is that the overall utility function is the sum of local utilities in all considered segments of the vehicle path. This is the case not only when the goal is to minimize the total time, but also when we aim at the minimal total fuel consumption or a linear combination of these two. The second key property is the Markov property. This allows us to aggregate the whole future in one probability and one utility potential. These potentials are defined over the speed variable in the current path segment.

☆ This work was supported by the Czech Science Foundation (project 16-12010S).
* Corresponding author.
E-mail addresses: velorex@utia.cas.cz (V. Kratochvíl), vomlel@utia.cas.cz (J. Vomlel).

http://dx.doi.org/10.1016/j.ijar.2016.11.018
0888-613X/

We illustrate the proposed method using an example of the speed profile optimization of a Formula 1 race car at the Silverstone F1 circuit [24,26]. The goal is to minimize the total lap time. This example will be used throughout the paper to explain the key concepts and for the final experimental evaluation of the proposed approach. An advantage of this example is that the optimal solution is known [24]. This allows us to compare both the influence diagram solution and the solution found by an optimal control theory method with the analytic solution.

In Section 2 we introduce the physical model of the vehicle using an ordinary differential equation. Since all constraints are naturally given with respect to the vehicle position, we describe the vehicle dynamics with respect to its position. The state variable will be the vehicle speed. In Section 3 we specify the optimal control problem we want to solve, which is to find a vehicle speed profile that minimizes the total time and satisfies all speed constraints.

In Section 5 we describe influence diagrams that can be used for this task. We explain the basic operations with probability and utility potentials. We also discuss the two methods that we use for the inclusion of the speed constraints. In Section 6 we derive an exact optimal algorithm for the continuous influence diagram. It is possible to find the exact solution due to the very specific form of the problem. In a more general setup such a solution would be hard to find. Therefore it is worthwhile to explore the behavior of the discretized version, which we do in Section 7. In this Section we introduce an enhancement used for the values that are on the border of the set of admissible speeds. Since similar problems are often solved using the methods of Nonlinear Optimal Control we briefly discuss the ACADO toolkit, which we use in the experiments to provide us a solution using a common method of optimal control – the multiple shooting method.

The final part of the paper is devoted to numerical experiments. We compare the solutions provided by influence diagrams with different discretizations, and with or without proposed enhancements, with the gold standard, which is the solution provided by the optimal algorithm we propose in Section 6. We also present comparisons with the results of ACADO and with real test pilots at the Silverstone F1 circuit. The experiments show that influence diagrams with the proposed enhancements can be very fast in providing good results that are close to the optimum. Each such solution can be used as a starting point of an optimal control algorithm that can further improve the precision of the solution.

2. The physical model of the vehicle

In this paper, the path of the vehicle will be fixed and known in advance. Therefore it is admissible to define the vehicle position as a distance from the start point. Let $s(t)$ be the vehicle position at time t. The speed at time t is the first derivative of the position with respect to time $v(t) = \dot{s}(t)$ and the acceleration at time t is the first derivative of the speed with respect to time $a(t) = \dot{v}(t) = \ddot{s}(t)$.

All model constraints will be given with respect to the vehicle position; therefore we describe the model dynamics with respect to the position. The state variable will be the speed. By the chain rule for the derivative of a composed function we have

$$a = \frac{dv}{dt} = \frac{dv}{ds} \cdot \frac{ds}{dt} = \frac{dv}{ds} \cdot v \ . \tag{1}$$

From this we get the following ordinary differential equation (ODE):

$$v \cdot \frac{dv}{ds} = a \ . \tag{2}$$

Let a^{max} be the maximum tangential engine acceleration of the vehicle[1] and a^{min} be the maximum tangential brakes deceleration.[2] Engine acceleration corresponding to position s is defined by the following equation:

$$a^e(u) = \begin{cases} u \cdot a^{max} & \text{if } u > 0 \\ u \cdot a^{min} & \text{otherwise,} \end{cases} \tag{3}$$

where u is a value of the control.[3] Negative values correspond to braking and positive values to accelerating. Let position s be from interval $[0, S]$, where $S \in \mathbb{R}$. The control function u is restricted by

$$-1 \leq u(s) \leq +1 \ . \tag{4}$$

Deceleration caused by aerodynamic drag is

$$a^d(s) = c_v \cdot v(s)^2 \ . \tag{5}$$

The actual vehicle acceleration at position s is

[1] Note that this is a property of the vehicle engine (without considering the aerodynamic drag and friction forces). The real maximum vehicle acceleration is lower.

[2] This is a property of the vehicle brakes (without considering the aerodynamic drag and friction forces). The real maximum deceleration is higher.

[3] We use the letter u because it is standard in control theory.

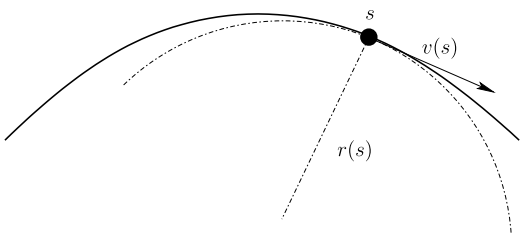

Fig. 1. A point mass moving along a path.

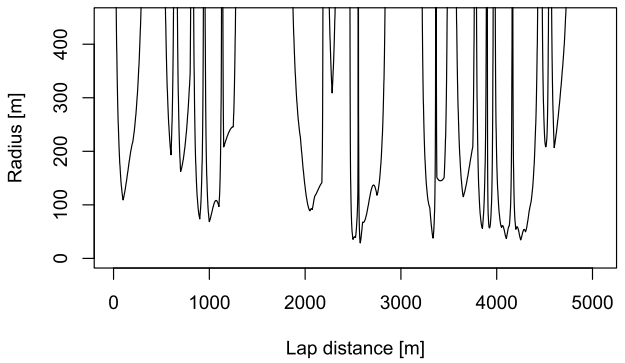

Fig. 2. The radius profile.

$$a(s) = a^e(u(s)) - a^d(s) - g \cdot \sin\beta(s) \ , \tag{6}$$

where $\beta(s)$ is the inclination angle at position s and g is the gravitational constant. Theoretically, $\sin\beta$ can range from -1 to $+1$, but for regular roads we can expect $\sin\beta \in [-0.2, +0.2]$. By substituting (6) into (2) we get the following equation that fully describes the vehicle dynamics:

$$v(s) \cdot \frac{dv(s)}{ds} = a^e(u(s)) - c_v \cdot v(s)^2 - g \cdot \sin\beta(s) \tag{7}$$

Example 1. All examples in this paper will follow [24] and we consider a Formula 1 race car with the engine acceleration characteristics $a^{max} = 16\ \mathrm{m\,s^{-2}}$ and $a^{min} = 18\ \mathrm{m\,s^{-2}}$. The constant characterizing aerodynamic drag will be $c_v = 0.0021\ \mathrm{m^{-1}}$.

3. Problem specification

The vehicle path is specified by a radius profile, which is defined as the radius $r(s)$ of the circular arc which best approximates the path curve at position s (see Fig. 1). The radius $r(s)$ defines the maximum speed at position s as

$$v(s) \le v^{max}(s) = \sqrt{a_n^{max} \cdot r(s)} \ , \tag{8}$$

where a_n^{max} is the maximum lateral acceleration, which is a vehicle property mainly influenced by the tires and by the aerodynamic downforce generated by the wings and by the low air pressure under the car. Following [24], we assume that the latter does not depend on the speed.

Example 2. For a typical F1 race car, $a_n^{max} = 30\ \mathrm{m\,s^{-2}}$. This implies that

$$v^{max}(s) = \sqrt{30 \cdot r(s)} \ . \tag{9}$$

If $r(s) = 30$ m then the maximum speed is 108 km/h.

Example 3. In Fig. 2, we present the radius profile of the F1 Silverstone circuit (the bridge version). Radii larger than 500 m are not depicted.[4] From the radius profile, we derive the maximum speed profile by means of Formula (9). See Fig. 3 for the maximum speed profile.

[4] A radius of 500 m allows for the maximum speed of 441 km/h – a speed never reached by an F1 race car.

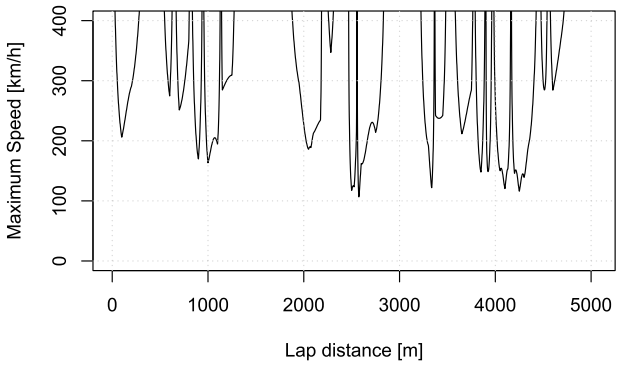

Fig. 3. The maximum speed profile.

We assume the initial speed to be given; thus we have the boundary condition $v(0) = v_0$, where v_0 is a constant known in advance.[5] To compute time $t(s, s+\Delta s)$ needed to get from position s to position $s+\Delta s$ we have to solve

$$\frac{ds}{dt} = v(s) \ , \tag{10}$$

which results in

$$t(s, s+\Delta s) = \int_{s}^{s+\Delta s} \frac{1}{v(s')} ds' \ . \tag{11}$$

The optimality criteria is to minimize the total time $t(S)$ taken to get to the goal at position $s = S$. We will assume $\beta(s) = 0$. We can specify the problem using the language of optimal control theory as:

$$\begin{aligned} &\underset{v(s)}{\text{minimize}} \quad \int_0^S \frac{1}{v(s)} ds \\ &\text{subject to} \\ &\frac{dv(s)}{ds} \cdot v(s) = a^e(u(s)) - c_v \cdot v(s)^2 \qquad \text{for all } s \in [0, S] \\ &0 < v(s) \leq v^{max}(s) \qquad \text{for all } s \in [0, S] \\ &-1 \leq u(s) \leq 1 \qquad \text{for all } s \in [0, S] \\ &v(0) = v_0 \ . \end{aligned} \tag{12}$$

4. Solution of the ODE for the vehicle speed and the total time

In the rest of the paper we will assume there is no inclination of the road, i.e., $\beta(s) = 0$ for all $s \in [0, S]$. By solving the ODE (7), we get the formula for the speed at position $s+\Delta s$

$$v(s+\Delta s) = \sqrt{v(s)^2 \exp(-2c_v \Delta s) \ + \ \frac{a^e(u(s))}{c_v} (1 - \exp(-2c_v \Delta s))} \ . \tag{13}$$

This function is illustrated in Fig. 4 where the speed is plotted as a function of the position for the control $u = 1$ (full throttle) and the initial speed of 0.1 km/h, and for the control $u = -1$ (maximum breaking) and the initial speed of 300 km/h, respectively.

We will also make use of the inverse function of v, which provides the value of the speed at position s for a given value of speed $v(s+\Delta s)$ at position $s+\Delta s$:

$$w(s) = \sqrt{v(s+\Delta s)^2 \exp(2c_v \Delta s) \ + \ \frac{a^e(u(s))}{c_v} (1 - \exp(2c_v \Delta s))} \ . \tag{14}$$

[5] Note that the initial speed coincides with the final speed (of the previous lap).

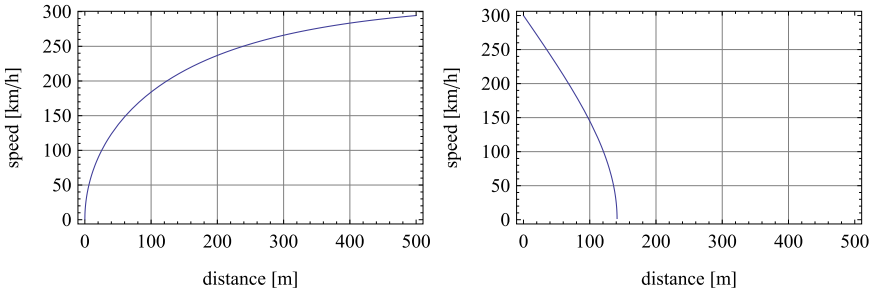

Fig. 4. The speed as a function of position for the control $u = 1$ and the initial speed of 0.1 km/h (left), and for the control $u = -1$ and the initial speed of 300 km/h (right).

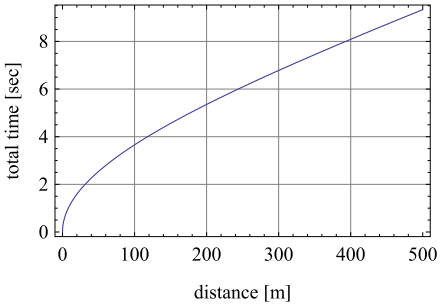

Fig. 5. The total time as a function of position for the control $u = 1$ and the initial speed of 0.1 km/h.

To derive the formula for the total time spent to get from position s to position $s + \Delta s$ we substitute (13) into (11) and solve the integral. We get the following Formula[6]

$$t(s, s+\Delta s) = \begin{cases} \dfrac{1}{\sqrt{a^e(u(s)) \cdot c_v}} \cdot \left(\begin{array}{l} \log\left(\dfrac{\sqrt{a^e(u(s)) \cdot \exp(2c_v \cdot \Delta s)}}{+\sqrt{a^e(u(s)) \cdot \exp(2c_v \cdot \Delta s) - a^e(u(s)) + c_v \cdot v(s)^2}} \right) \\ -\log\left(\sqrt{a^e(u(s))} + \sqrt{c_v \cdot v(s)^2} \right) \end{array} \right) & \text{if (c1)} \\ \dfrac{\exp(c_v \cdot \Delta s)}{c_v \cdot v(s)} & \text{if (c2)} \\ +\infty & \text{otherwise.} \end{cases} \tag{15}$$

We assume $v(s) \geq 0$, the condition (c1) is

$$\text{if } a^e(u(s)) > 0 \text{ or } \left(a^e(u(s)) < 0 \text{ and } v(s) \geq \sqrt{\frac{a^e(u(s))}{c_v} \cdot (1 - \exp(2c_v \cdot \Delta s))} \right),$$

the condition (c2) is

$$a^e(u(s)) = 0 \text{ and } v(s) > 0.$$

Since this Formula is quite complicated, we illustrate the function in Fig. 5 where the total time is plotted as a function of the position for the control $u = 1$ and the initial speed of 0.1 km/h.

Remark 1. If we further simplify the model by disregarding the decrease of acceleration due to the aerodynamic drag within the segment $[s, s + \Delta s]$ then the acceleration is constant and equal to $a(s) = a^e(u(s)) - c_v \cdot v(s)^2$ at the segment $[s, s + \Delta s]$. Then we get the Formula

$$v(s + \Delta s) = \sqrt{v(s)^2 \ + \ 2 \cdot a(s) \cdot \Delta s} \ , \tag{16}$$

which corresponds to the well-known formula of the motion with the uniform acceleration $a(s)$. This is the Formula used in the experiments reported in [11]. In this paper we stick to Formula (13) since it is more precise. In Fig. 6, we compare speed profiles generated by Formulae (13) (full line) and (16) (dashed line), using constant control $u = 1$ with the segment

[6] Note that during the computations using this Formula we may get complex numbers but the result is always a real number.

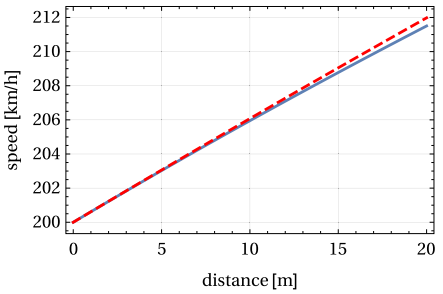

Fig. 6. The speed profiles for the control $u = 1$ and the initial speed of 200 km/h at a path segment 20 m long for the full and simplified speed formulae.

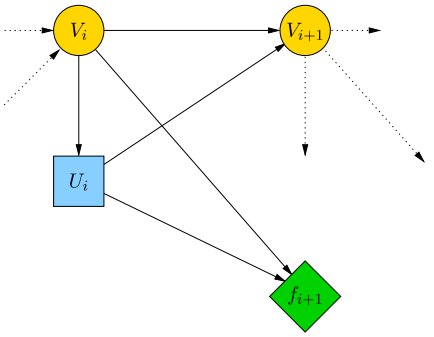

Fig. 7. A part of the influence diagram for one path segment.

initial speed of $v(s) = 200$ km/h at a path segment 20 m long. Note that for the initial speed of 200 km/h, the speed difference after 20 m is about 0.48 km/h.

5. Influence diagrams for vehicle speed optimization

An influence diagram [7] is a Bayesian network augmented with decision variables and utility functions. In this section, we specify an influence diagram for the problem of optimization of a vehicle speed profile [11].

We split the vehicle path into n segments of the same length Δs. Let $[s, s+\Delta s]$ denote the segment starting at position s. We assume the control $u(s)$ (and therefore the engine acceleration $a^e(s)$ – as defined in (3)) to be constant at the segment $[s, s+\Delta s]$. For $i = 0, 1, \ldots, n-1$ corresponding to the positions $i \cdot \Delta s$ of the vehicle there is:

- one random variable V_i representing the speed and taking values from a set $\mathcal{V}$,
- one decision variable U_i representing the control and taking values from a set $\mathcal{U}$,
- one utility potential f_{i+1} representing the time to drive a path segment and taking values from $\mathbb{R}^+$, and

one random variable V_n for the final speed taking values from a set $\mathcal{V}$. The variables U_i and V_i will be either continuous (Subsection 5.1) or discrete (Subsection 5.2).

In Fig. 7, we present the structure of a part of the influence diagram corresponding to one segment of the path.

The physical model of the vehicle (given in Sections 2 and 4) is used to define the probability and utility functions of the influence diagram. First, we define the function that provides the speed at the end of a segment i with its arguments being the speed v_i at the beginning of the segment and the control u_i in the segment.

Definition 1 *(Speed at the end of a path segment).* Let u_i be the value of the control variable in the path segment $[i \cdot \Delta s, (i+1) \cdot \Delta s]$ and v_i the speed at the beginning of this segment. We define function v' that provides the speed at the end of this segment by

$$v'(u_i, v_i) = v((i+1) \cdot \Delta s) \ , \tag{17}$$

where the function v is defined by (13) with the substituted values of $v(s) = v_i$ and $u(s) = u_i$.

In Section 6 we will use the inverse function of v'.

Definition 2 *(Speed at the beginning of a path segment).* We define function w' that gives the initial speed v_i such that after driving distance Δs with the control u_i the speed is v_{i+1} as

$$w'(u_i, v_{i+1}) = w(i \cdot \Delta s) \ , \tag{18}$$

where the function w is defined by (14) with the substituted values of $v(s) = v_i$ and $u(s) = u_i$.

Definition 3 *(Time to drive a path segment).* We define function f that provides the time to drive the path segment $[i \cdot \Delta s, (i+1) \cdot \Delta s]$ as a function of the initial speed v_i and the control u_i

$$f(u_i, v_i) = t(i \cdot \Delta s, \ (i+1) \cdot \Delta s) \ , \tag{19}$$

where the function t is defined by (15) with the substituted values of $v(s) = v_i$ and $u(s) = u_i$.

This function defines the values of the utility potentials in both types of the influence diagrams (i.e., continuous and also discrete ones).

5.1. Influence diagram with continuous variables

Let $v'(u_i, v_i)$ be the function specified in (17). For $i = 0, \ldots, n-1$ and all combinations of values (u_i, v_i, v_{i+1}) we define the conditional probability

$$P(V_{i+1} = v_{i+1} | U_i = u_i, V_i = v_i) = \begin{cases} 1 & \text{if } v_{i+1} = v'(u_i, v_i) \\ 0 & \text{otherwise.} \end{cases}$$

In this case the vehicle behavior is deterministic.

5.2. Influence diagrams with discrete variables

It is often not possible to find a precise solution to the influence diagrams with continuous variables (especially when utilities are nonlinear). In such a case, a natural option is to work with discrete variables. In the problem of speed profile optimization, we will discretize the speed and control variables equidistantly, i.e., the sets $\mathcal{V}, \mathcal{U}$ will be finite with the discretization steps being d_V, d_U, respectively.

However, this brings a fundamental problem since, despite $v_i \in \mathcal{V}$ and $u_i \in \mathcal{U}$ then, typically, the value $v'(u_i, v_i)$ of variable V_{i+1} defined by Formula (17) is not in $\mathcal{V}$. For this reason we approximate

$$P(V_{i+1} = v'(u_i, v_i) | U_i = u_i, V_i = v_i) = 1$$

by two conditional probabilities

$$P(V_{i+1} = \underline{v}_{i+1} | U_i = u_i, V_i = v_i) \ \text{ and } \ P(V_{i+1} = \overline{v}_{i+1} | U_i = u_i, V_i = v_i)$$

for values $\underline{v}_{i+1} \leq v'(u_i, v_i)$ and $\overline{v}_{i+1} \geq v'(u_i, v_i)$ that are values from $\mathcal{V}$ closest to $v'(u_i, v_i)$. We define the probabilities in the following way

$$P(V_{i+1} = v | U_i = u_i, V_i = v_i) = \begin{cases} 1 - \dfrac{|v - v'(u_i, v_i)|}{d_V} & \text{if } v \in \{\underline{v}_{i+1}, \overline{v}_{i+1}\} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

The idea behind Formula (20) is that the closer the value of $\underline{v}_{i+1}, \overline{v}_{i+1} \in \mathcal{V}$ to the exact speed value $v'(u_i, v_i)$ the greater its probability. The definition given above guarantees the expected value of V_{i+1} given $U_i = u_i$ and $V_i = v_i$ is equal to $v'(u_i, v_i)$ defined by Formula (17). We will illustrate the idea in the following example.

Example 4. Assume a speed of $v_i = 120$ km/h and four different values of u^1, u^2, u^3, u^4 of U_i for which Formula (17) gives speed values 126, 129, 137, and 145 (in km/h), respectively. Let $\mathcal{V} \supset \mathcal{V}_i = \{120, 130, 140, 150\}$ km/h.

Then the vectors $(p_v, v \in \mathcal{V}_i)$ of conditional probabilities $P(V_{i+1} = v | U_i = u_i, V_i = v_i)$ are defined as

$$P(V_{i+1} = v_{i+1} | V_i = v_i, U_i = u_i) = \begin{cases} (0.4, 0.6, 0, 0) & \text{for } u_i = u^1 \\ (0.1, 0.9, 0, 0) & \text{for } u_i = u^2 \\ (0, 0.3, 0.7, 0) & \text{for } u_i = u^3 \\ (0, 0, 0.5, 0.5) & \text{for } u_i = u^4 \end{cases}$$

See Fig. 8 for the visualization of the above-specified conditional probabilities.

The expected values of V_{i+1} are

$$E(V_{i+1} | V_i = v_i, U_i = u_i) = \begin{cases} 0.4 \cdot 120 + 0.6 \cdot 130 \ = \ 126 & \text{for } u_i = u^1 \\ 0.1 \cdot 120 + 0.9 \cdot 130 \ = \ 129 & \text{for } u_i = u^2 \\ 0.3 \cdot 130 + 0.7 \cdot 140 \ = \ 137 & \text{for } u_i = u^3 \\ 0.5 \cdot 140 + 0.5 \cdot 150 \ = \ 145 & \text{for } u_i = u^4 \end{cases} ,$$

which are the values of $v'(u_i, v_i)$ defined by Formula (17).

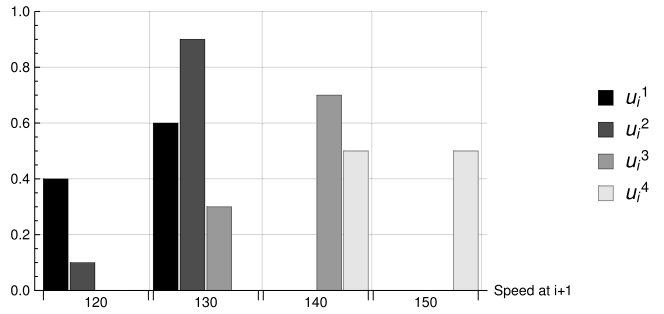

Fig. 8. Conditional probabilities of speed values of V_{i+1} for four different control values $u_i^1, u_i^2, u_i^3, u_i^4$.

By the above-described approximation the relations between variables U_i, V_i, and V_{i+1} are no more deterministic. We approximated the deterministic continuous influence diagram by a non-deterministic discrete influence diagram.

Remark 2. The discretization also causes problems during the inference around the values v_i^{max} and v_i^{min}. We will address this issue in Section 7.

5.3. Operations with probability and utility potentials

We will use Greek alphabet symbols to denote probability and utility potentials. φ is a probability potential if it maps from the Cartesian product of state spaces of its variables to the interval $[0, 1]$ and ψ a utility potential if it maps from the Cartesian product of state spaces of its variables to real numbers $\mathbb{R}$. From this point forward, we will use the following abbreviations

$$\sum_{V_i} \varphi(V_i, \cdot) = \sum_{v_i \in \mathcal{V}_i} \varphi(V_i = v_i, \cdot) \text{ and}$$
$$\max_{U_i} \psi(U_i, V_i) = \max_{u_i \in \mathcal{U}_i(v_i)} \psi(U_i = u_i, V_i = v_i) \ .$$

$\mathcal{M}$ will be a generalized marginalization operation. The operator $\mathcal{M}$ acts differently for a discrete random variable A, a continuous random variable B, and a decision variable U of a (probability or utility) potential ξ:

$$\underset{A}{\mathcal{M}}\, \xi(A, \ldots) = \sum_{A} \xi(A, \ldots),$$
$$\underset{B}{\mathcal{M}}\, \xi(B, \ldots) = \int \xi(B = b, \ldots)\, db,$$
$$\underset{U}{\mathcal{M}}\, \xi(U, \ldots) = \max_{U} \xi(U, \ldots) \ .$$

For a set of variables C, we define $\mathcal{M}_C\ \xi(C, \ldots)$ as a sequence of single-variable marginalizations.

5.4. Speed constraints as admissible control sets

A solution of the optimal speed profile problem has to fulfill constraints on the speed. We consider two basic ways of implementing the speed constraints in the inference algorithm, which we are going to discuss in this and the next Sections.

We assume that a maximum speed v_i^{max} and a minimum speed v_i^{min} are given in advance at each path coordinate $i = 1, \ldots, n$. Let the admissible set of speed values at the end of the path be

$$\mathcal{V}_n = \{v \in \mathcal{V}, v_n^{min} \leq v \leq v_n^{max}\} \ . \tag{21}$$

Now, for $i = n-1, n-2, \ldots, 1$ we will inductively apply the speed constraints so that we allow only those control signals $u_i \in \mathcal{U}$ that lead to $v_{i+1} = v'(u_i, v_i)$ for which $v_{i+1} \in \mathcal{V}_{i+1}$. It means that we define functions $\mathcal{U}_i(V_i)$ such that, for each value v_i of variable V_i, they provide the set of admissible control values:

$$\mathcal{U}_i(v_i) = \{u \in \mathcal{U} : v'(u_i, v_i) \in \mathcal{V}_{i+1}\} \ . \tag{22}$$

This set inductively defines the set of admissible speed values at i for which there exists an admissible control value:

$$\mathcal{V}_i = \{v \in \mathcal{V} : v_i^{min} \leq v \leq v_i^{max},\ \mathcal{U}_i(v) \neq \emptyset\} \ . \tag{23}$$

This, again, inductively defines a set $\mathcal{U}_{i-1}(v_{i-1})$. This process is repeated until $i = 1$.

Remark 3. If the speed variables $V_i, i = 0, 1, \ldots, n$ are continuous then in each step i we would need to derive functions $\mathcal{U}_i : \mathcal{V} \to \mathcal{U}$. However, as we will see in Section 6, a simpler approach is available due to a specific nature of our problem.

5.5. Speed constraints as likelihood evidence

Likelihood evidence is a vector that takes a value between zero and one for each state of the corresponding variable [8, Section 1.4.6]. In the discrete case, the likelihood evidence is, for a given speed constraint for variable V_i, a vector $\varphi_i^e(v_i^{min}, v_i^{max})$ of length $|\mathcal{V}|$ such that

$$\varphi_i^e(v_i^{min}, v_i^{max}) = \begin{cases} 1 & \text{if } v_i^{min} \le v_i \le v_i^{max} \\ 1 - \dfrac{v_i - v_i^{max}}{d_V} & \text{if } v_i = \min\{v \in \mathcal{V}, v > v_i^{max}\} \\ 1 - \dfrac{v_i^{min} - v_j}{d_V} & \text{if } v_j = \max\{v \in \mathcal{V}, v < v_i^{min}\} \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

The idea behind Formula (24) is that the closer the value of v_i^{max} is to the nearest speed value $v \in \mathcal{V}$ that is greater than v_i^{max}, the higher is the likelihood of v. A similar argument applies to v_i^{min}. During the inference, we propagate the constraints using the probability potentials.

$$\varphi_i(V_i) = \mathcal{M}_{V_{i+1},U_i}\, P(V_{i+1}|U_i, V_i) \cdot \varphi_{i+1}^e(0, v_{i+1}^{max}) \ ,$$

where φ_i is the probability potential sent to the neighboring clique. See the following example that illustrates this process.

Example 5. Let $\mathcal{V} = \{0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ m/s be the set of speed values and $\mathcal{U} = \{-1, -0.8, -0.6, -0.4, -0.2, 0.2, 0.4, 0.6, 0.8, 1\}$ the set of control values. Assume $\Delta s = 1$ and $v_2^{max} = 42$ m/s. Following (24) we set the likelihood evidence

$$\varphi_2^e(0, v_2^{max}) = (1, 1, 1, 1, 1, 0.2, 0, 0, 0, 0, 0) \ .$$

Next, we calculate the 3-dimensional probability potential characterizing the vehicle speed $P(V_2|U_1, V_1)$ using (20) and the car characteristics from Example 1. Since it has $|\mathcal{U}| \cdot |\mathcal{V}|^2$ values, which in this example is 11^3, we present only a few values as an example:

$$P(V_2|U_1 = u_1, V_1 = v_1) = \begin{cases} (0.202, 0.798, 0, 0, 0, 0, 0, 0, 0, 0, 0) & \text{for } u_1 = -1 \text{ and } v_1 = 10 \text{ m/s} \\ (0.002, 0.998, 0, 0, 0, 0, 0, 0, 0, 0, 0) & \text{for } u_1 = 0 \text{ and } v_1 = 10 \text{ m/s} \\ (0, 0.853, 0.147, 0, 0, 0, 0, 0, 0, 0, 0) & \text{for } u_1 = 1 \text{ and } v_1 = 10 \text{ m/s} \\ \ldots & \\ (0, 0, 0, 0, 0, 0, 0.040, 0.960, 0, 0, 0) & \text{for } u_1 = -1 \text{ and } v_1 = 70 \text{ m/s} \\ (0, 0, 0, 0, 0, 0, 0.015, 0.985, 0, 0, 0) & \text{for } u_1 = 0 \text{ and } v_1 = 70 \text{ m/s} \\ (0, 0, 0, 0, 0, 0, 0, 0.992, 0.008, 0, 0) & \text{for } u_1 = 1 \text{ and } v_1 = 70 \text{ m/s} \\ \ldots & \end{cases}$$

When marginalizing we first sum over the values of V_2 and then take maximum over the values of U_1, i.e., we get

$$\mathcal{M}_{V_2,U_1}\Big(P(V_2|U_1, V_1) \cdot \varphi_2^e(0, 42)\Big) = (1, 1, 1, 1, 1, 0.237, 0.009, 0, 0, 0, 0) \ .$$

Note that this approximately corresponds to the likelihood evidence

$$\varphi_1^e(0, 42.37) = (1, 1, 1, 1, 1, 0.237, 0, 0, 0, 0, 0)$$

which means that for the speed $v_1 = 42.37$ m/s there exists a control $u_1 \in \mathcal{U}$ such that we can get to a speed value lower than the required 42 m/s. Actually, it holds that $v'(-1, 42.37) = 41.88 < 42$.

5.6. Policy and strategy

The control of the vehicle speed is realized by means of the control policy.

Definition 4. Control policy is a set of functions

$$\delta = \left\{\delta(U_i|V_i) : \ i \in \{1, \ldots, n-1\}, v_i \in \mathcal{V}\right\}$$

such that for all $i = 1, \ldots, n$ and all $v_i \in \mathcal{V}$ it maps $u_i \in \mathcal{U}$ to values from $[0, 1]$ and it holds that

$$\sum_{u_i \in \mathcal{U}} \delta(U_i = u_i|V_i = v_i) = 1 \ . \tag{25}$$

Definition 5. A control policy δ is deterministic if for all $i = 1, \ldots, n$ and all $v_i \in \mathcal{V}$ it holds that there is a function $\delta_i : \mathcal{V} \to \mathcal{U}$ such that for all $u \in \mathcal{U}$

$$\delta(U_i = u|V_i = v_i) = \begin{cases} 1 & \text{if } u = \delta_i(v_i) \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

Remark 4. In this paper, all considered policies will be deterministic.

Definition 6. The expected value E_f of a deterministic control policy δ specified by functions u_i is the sum or the integral over all possible configurations of random variables corresponding to the products of the probability and the criteria values of that configuration:

$$E_f(\delta) = \underset{V_1,\ldots,V_n}{\mathcal{M}} P(V_1, \ldots, V_n) \cdot f(V_1, \ldots, V_n) \tag{27}$$

where

$$P(V_1, \ldots, V_n) = P(V_1) \cdot \prod_{i=1}^{n-1} P(V_{i+1}|U_i = u_i(v_i), V_i) \tag{28}$$

$$f(V_1, \ldots, V_n) = \sum_{i=1}^{n-1} f(V_i, V_{i+1}) \ . \tag{29}$$

The criteria to be optimized will be the expected value E_f of a deterministic control policy.

Definition 7. An optimal deterministic policy δ^* is a deterministic policy such that

$$E_f(\delta) \leq E_f(\delta^*) \tag{30}$$

holds for all control policies δ. We will use the symbol u_i^* to denote the function $u_i : \mathcal{V} \to \mathcal{U}$ that specifies the optimal deterministic policy δ^* according to Definition 5. The symbol $u_i^*(V_i)$ denotes the set of all functions u_i^* for all values v_i of variable V_i.

5.7. Efficient computation of an optimal control policy

Using the recursive application of the commutative and distributive laws, we get the following theorem that specifies a computationally efficient algorithm for finding an optimal decision policy. Note that our algorithm is just a special case of general inference methods for influence diagrams [9,21,20]. But since our influence diagram has a simple structure, it is useful to derive an inference algorithm tailored for the task we are solving. Note that, in this case, because of the topology of the influence diagram, the algorithm does not involve divisions. You can find more about that aspect in [22]. The computations can also be viewed as a special case of dynamic programming [2].

Theorem 1.

$$E_f^* = E_f(\delta^*) = \underset{V_1}{\mathcal{M}} P(V_1) \cdot \psi(V_1) \ , \tag{31}$$

where $\psi(V_1)$ is computed recursively for $i = 1, \ldots, n-1$ as

$$\psi(V_i) = \max_{U_i} \underset{V_{i+1}}{\mathcal{M}} P(V_{i+1}|V_i, U_i) \cdot \Big(f(V_i, V_{i+1}) + \psi(V_{i+1})\Big) \tag{32}$$

with the recursion terminal values being $\psi(V_n) = \mathbf{0}(V_n)$, where $\mathbf{0}(V_n)$ stands for the vector taking the zero value for all states of variable V_n.

The proof can be found in Appendix A.

Remark 5. In each step $i = 1, \ldots, n$, an optimal deterministic policy is specified (according to Definition 5) by a function $u_i : \mathcal{V} \to \mathcal{U}$ such that $u_i(v_i) = u_i^*(v_i)$, where $u_i^*(v_i)$ is the value of U_i that maximizes Formula (32) for a given v_i.

6. Continuous influence diagram solution

In this section, we present an algorithm that finds an optimal speed profile for a given discretization of the vehicle path. Recall that the optimality criterion is the total time and our goal is to minimize it.

Next we present a Corollary to Theorem 1 that specifies an algorithm for the case of deterministic vehicle behavior.

Corollary 1. *Assume that the vehicle behavior is deterministic. Then*

$$E_f^* = E_f(\delta^*) = \underset{V_1}{\mathcal{M}}\, P(V_1) \cdot \psi(V_1) \ , \tag{33}$$

where $\psi(V_1)$ is computed recursively for $i = 1, \ldots, n-1$ and for all $v_i \in \mathcal{V}$ as:

$$\psi(v_i) = f\left(v_i, v'\left(\max\, \mathcal{U}_i(v_i), v_i\right)\right) + \psi\left(v'\left(\max\, \mathcal{U}_i(v_i), v_i\right)\right) \ . \tag{34}$$

The recursion terminal values are defined as $\psi(v_n) = 0$ for all $v_n \in \mathcal{V}$.

Proof. Formula (34) follows from (32) – the optimization is given by minimizing the total time. Therefore $\max_{U_i}$ corresponds to picking the highest value from $\mathcal{U}_i(v_i)$. Also, note that for the deterministic vehicle behavior and for any potential $\xi(V_i, V_{i+1})$ it holds for all $u_i \in \mathcal{U}, v_i \in \mathcal{V}$ that

$$\underset{V_{i+1}}{\mathcal{M}}\, P(V_{i+1}|U_i = u_i, V_i = v_i) \cdot \xi(V_i = v_i, V_{i+1}) = \xi(V_i = v_i, V_{i+1} = v'(u_i, v_i)) \ . \quad \square$$

From Corollary 1 we derive a computationally efficient Algorithm 1 that can be used to efficiently compute the optimal speed profile of the vehicle satisfying the speed constraints. We will use the function $w(u_i, v_{i+1})$ that gives the initial speed v_i such that the speed is v_{i+1} after driving the distance s with the control u_i. The idea behind this algorithm is that the function f, which is to be maximized, implies that the best policy for any $v_i, i = 1, \ldots, n-1$ is to speed up as much as possible to be able to slow down by the maximum allowed deceleration to satisfy $v_j^* \le v_j^{max}$ for all $j > i$. First, the maximal speed profile is constructed from the speed constraints and the maximum deceleration of the vehicle. Second, the best policy is found with the maximum acceleration until the speed meets the maximum profile constructed at the first stage of the algorithm.

input : $v_i^{max}, i = 1, \ldots, n$ – maximal speed values
output: $v_i^*, i = 1, \ldots, n$ – speed values minimizing the total time

$v_n^* = v_n^{max}$;
for $i = n-1, \ldots, 1$ **do**
  $v_i^* = w'(-1, v_{i+1}^*)$;
  if $(v_i^* > v_i^{max})$ **then**
    $v_i^* = v_i^{max}$;
  end
end
for $i = 1, \ldots, n-1$ **do**
  $v_{i+1} = v'(+1, v_i^*)$;
  if $(v_{i+1} < v_{i+1}^*)$ **then**
    $v_{i+1}^* = v_{i+1}$;
  end
end

Algorithm 1: Optimal speed profile construction.

We should stress that it is possible to find the exact solution of the continuous influence diagram thanks to a very specific form of our problem. In a more general setup such a solution would be hard to find and quite often impossible without resorting to numeric approximation methods. Therefore it is worthwhile to explore the behavior of the discretized version, which we are going to do in the next Sections. The exact solution of the continuous influence diagram will be used as the gold standard for the comparisons.

7. Discrete influence diagram solution

Among the first influence diagram inference algorithms are those published in [19,21,9,12]. In some special cases – see [13] – the inference can be simplified so that no division operation is needed. This is also the case of the influence diagram for the optimal speed profile – as was shown in Theorem 1.

Let us briefly recall the solution algorithm from [9]: First, utility nodes are eliminated and the ID is converted into a junction tree. To every clique C in the junction tree, we associate a probability potential φ_C and a utility potential ψ_C. Cliques follow the information ordering. In the influence diagram for the speed profile optimization, the cliques are $C_i = \{U_i, V_i, V_{i+1}\}$, $i = 0, 1, \ldots, n-1$. In this case, the junction tree is initiated as follows: $\varphi_{C_i} = P(V_{i+1}|U_i, V_i)$, $\psi_{C_i} = f(U_i, V_i)$ as they specified by Formulae (20) and (19), respectively.

Let C_i and C_{i+1} be adjacent cliques with separator S_i. To pass a message from clique C_{i+1} to clique C_i potentials φ_{C_i} and ψ_{C_i} are updated[7] as follows [9]:

$$\psi'_{C_i} = \psi_{C_i} + \frac{\psi_{S_i}}{\varphi_{S_i}} \ , \tag{35}$$

$$\varphi'_{C_i} = \varphi_{C_i} \cdot \varphi_{S_i} \ , \tag{36}$$

where

$$\varphi_S = \underset{C_{i+1}\setminus S_i}{\mathcal{M}} \varphi_{C_{i+1}}, \quad \psi_{S_i} = \underset{C_{i+1}\setminus S_i}{\mathcal{M}} \left(\varphi_{C_{i+1}} \cdot \psi_{C_{i+1}}\right) \tag{37}$$

and $\mathcal{M}$ is a generalized marginalization operation as defined in Section 5.3.

The elimination order follows the inverse order as determined by a sequence that respects the information constraints in the sense that because V_k precedes V_{k+1} in the information sequence, V_{k+1} must be marginalized before V_k. Similarly, V_i precedes U_i and therefore U_i must be marginalized before V_i. In the process of marginalizing the decision (control) variable U_i in (37) to get ψ_S, we keep track of where the maximum is attained – it is a function of the remaining variables in the domain, which is V_i in our case. This yields a *policy* for the decision variable. The collection of all policies constitutes an *optimal strategy* for the ID. The optimal strategy together with the initial speed is used to determine the optimal speed and control profile. To start the marginalization process we initialize with $\psi_S = 0$ and $\varphi_S = 1$.

The analytical complexity of the algorithm is $O(|\mathcal{U}| \cdot |\mathcal{V}|^2 \cdot n)$.

7.1. Including speed constraints during the inference

The first method of incorporating speed constraints follows. At first, we represent all speed constraints as likelihood evidence on the speed as it is discussed in Section 5.5. Let $C_i = \{V_{i+1}, U_i, V_i\}$. During the updating by Formula (36) we include the respective speed constraint using evidence potential $\varphi^e_{i+1} = \varphi^e_{i+1}(v^{min}_{i+1}, v^{max}_{i+1})$ to the probability potential φ_{C_i} as follows:

$$\varphi'_{C_i} = \varphi_{C_i} \cdot min(\varphi_{S_i}, \varphi^e_{i+1}) \ . \tag{38}$$

The rest of the algorithm remains the same.

Remark 6. Note that this way of including speed constraints is heuristics only. However, it can be supported by the fact that φ_S passing through the separator can also be viewed as likelihood evidence on possible speed values (see Example 5). By using the minimum operator in (38) we consider only more restrictive values from the two potentials.

7.2. Speed constraints – control

In the second approach, we rewrite speed constraints into control constraints following Section 5.4. First, forbidden combinations (u_i, v_i) are identified. To do so, a zero-one table $T(U_i, V_i)$ is created:

$$T(U_i = u_i, V_i = v_i) = \begin{cases} 1 & \text{if } v'(u_i, v_i) \le v^{max}_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

Then, we eliminate the forbidden combinations (u_i, v_i) from the maximization process (37). When computing the policy for $V_i = v_i$ during elimination of the decision variable U_i, only the maximum on $\{u \in \mathcal{U} : T(V_i = v_i, U_i = u) = 1\}$ is taken into account. If $\forall u \in \mathcal{U}; T(V_i = v_i, U = u) = 0$ we put $\delta_i(v_i) = -1$ and $\psi_S(v_i) = 0$. That is,

$$\psi_S(V_i) = \max_{U_i} (\psi(U_i, V_i) \cdot T(U_i, V_i)) \ ,$$

[7] Given two potentials φ and ψ, their product $\varphi \cdot \psi$ and the quotient φ/ψ are defined in the natural way, except that $0/0$ is defined to be 0 and $x/0$ is undefined for $x \neq 0$ [9].

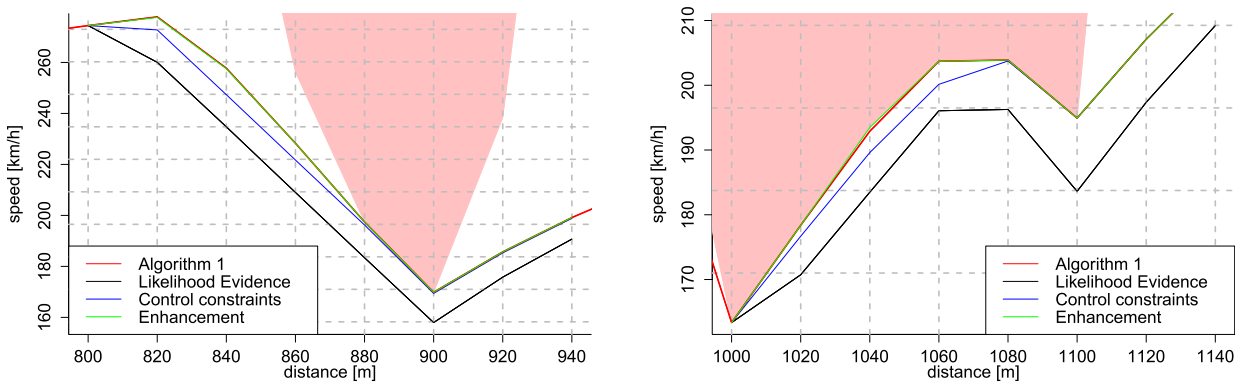

Fig. 9. Comparison of speed restriction implementations. The shaded areas correspond to the speed values forbidden by the constraints. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

where

$$\psi(U_i, V_i) = \sum_{V_{i+1}} \varphi_{C_i} \cdot \psi_{C_i}$$

for $C_i = \{V_{i+1}, U_i, V_i\}$. The difference of the above-mentioned approaches is illustrated by Fig. 9 where short parts of the constructed vehicle speed profiles are depicted.

7.3. Computational enhancements for the border speed values

Let $\mathcal{V}_i, i = 0, 1, \ldots, n$ be the set of admissible speed values computed as specified in Section 5.4. We will refer to the minimum and maximum values from these sets as to the *border speed* values. As we saw in Algorithm 1, the optimal solutions of our problem consist of border speed values only. Therefore it is of a special importance to be as precise as possible in the computations close to the border. The precision can be significantly improved if the utility for speed values just beyond the border speed values are also computed. The idea is simple. First, we will illustrate the problem we want to address by the following example.

Example 6. Let V_i and V_{i+1} be two consecutive speed variables. Further let the maximum allowed speed at i and $i+1$ be $v_i^{max} = 31$ and $v_{i+1}^{max} = 32$, respectively.[8] Assume that we can derive

$$\begin{aligned} v'(+0.1, 32) &= 32 \ \leq v_{i+1}^{max} \\ v'(+1.0, 31) &= 32 \ \leq v_{i+1}^{max} \\ v'(+1.0, 30) &= 31 \ \leq v_{i+1}^{max} \end{aligned}$$

using (17) representing the vehicle characteristics. If the best strategy is to drive as fast as possible then the best policies $\delta_i(v_i)$ for given speed values v_i are

$$\begin{aligned} \delta_i(32) &= +0.1 \\ \delta_i(31) &= +1.0 \\ \delta_i(30) &= +1.0 \ . \end{aligned}$$

Now, assume that the speed values are discretized with a discretization step of $d_V = 2$ so that $\mathcal{V} = \{\ldots, 30, 32, 34, \ldots\}$ and the considered speed at i is $v_i = 31$. Since $v_i = 31 \notin \mathcal{V}$ we have to estimate the value of $\delta_i(31)$ from the values of $\mathcal{V}$ that are the closest to $v_i = 31$, that is, 30 and 32. In this case we get:

$$\delta_i(31) = 0.5 \cdot \delta_i(30) + 0.5 \cdot \delta_i(32) = +0.55 \ .$$

We can see that this value is quite distant from the optimal control for $v_i = 31$, which is $+1.0$. This implies that the recommended vehicle speed values keep a certain distance from the speed limits. This behavior is illustrated by the blue line in Fig. 9b at the distance $s \in [1000, 1100]$. The plot grid corresponds to the speed and track discretizations.

[8] In this example we provide the speed values in m/s.

The behavior – illustrated by Example 6 – can be suppressed by a finer speed discretization at the expense of a higher computational complexity. A less computationally demanding possibility is an enhancement based on the following idea.[9] We can derive the border speed from the potential φ_{S_i} computed during the propagation. During the inference, we derive the value of the maximum speed v^M with a non-zero probability (the border speed) using the inverse of (24) applied to the result of the combination operation specified in Formula (38):

$$\min(\varphi_{S_i}, \varphi_i^e(0, v_i^{max})) \ . \tag{39}$$

We will use the following definition where we define two values from $\mathcal{V}$ that are the nearest to a given speed value v.

Definition 8. Let $\mathcal{V}$ be a discrete set of speed values and $v \in \mathbb{R}^+$ be a speed value such that $\min \mathcal{V} \leq v \leq \max \mathcal{V}$. Then

$$\lfloor v \rfloor_{\mathcal{V}} = \max\{v' \in \mathcal{V}, v' \leq v\}$$
$$\lceil v \rceil_{\mathcal{V}} = \min\{v' \in \mathcal{V}, v' \geq v\} \ .$$

The symbol $\mathcal{V}$ may be omitted if the context is clear.

First, we compute the policy δ_i for the decision variable U_i given the speed v for all speed values $v \in \mathcal{V}$ plus for the value v^M, which is the border speed value defined by (39). Second, we check whether $\delta_i(\lfloor v^M \rfloor) \neq \delta_i(\lceil v_M \rceil)$. Assume that it is the case.[10] Now, the goal is to modify the policy for $\delta_i(\lceil v_M \rceil)$ so that the weighted average of policies $\delta_i(\lceil v_M \rceil)$ and $\delta_i(\lfloor v_M \rfloor)$ equals to the desired $\delta_i(v_M)$:

$$(1-\alpha) \cdot \delta_i(\lceil v_M \rceil) + \alpha \cdot \delta_i(\lfloor v_M \rfloor) = \delta_i(v_M) \ , \tag{40}$$

where

$$\alpha = \frac{\lceil v^M \rceil - v^M}{d_V} \ . \tag{41}$$

This is required because Formula (40) specifies how the value of $\delta_i(v_M)$ is computed during the inference. From (40) we get that

$$\delta_i(\lceil v_M \rceil) = \frac{\delta_i(v_M) - \alpha \cdot \delta_i(\lfloor v_M \rfloor)}{(1-\alpha)} \ . \tag{42}$$

Note that the value of $\delta_i(\lceil v_M \rceil)$ may lay outside of the interval $[-1, +1]$. This does not cause any problems since this policy is never used directly, it is only used for the computation of the policy for the border speed value.

Using a similar line of reasoning, we derive the formula for the value of the utility potential $\psi_{S_i}(\lceil v_M \rceil)$:

$$\psi_{S_i}(\lceil v_M \rceil) = \frac{\psi_{S_i}(v_M) - \alpha \cdot \psi_{S_i}(\lfloor v_M \rfloor)}{(1-\alpha)} \ . \tag{43}$$

To highlight the differences between all approaches to implementing speed constraints, we choose a low discretization of speed and track $|\mathcal{V}| = 25$, $\Delta s = 20$ m. We calculate speed profiles in two parts of the track and depict them in Fig. 9. The grid corresponds to the chosen discretizations of speed and track. With a finer discretization, this behavior tends to disappear. Note that with the implemented enhancements, the profile overlaps the solution given by Algorithm 1 (for the respective track discretization).

7.4. Zero compression method

To speed up the computations in influence diagrams we employ the property that all probability potentials $P = P(V_{i+1}|U_i, V_i)$ are not only the same but also very sparse since for each configuration of parents they represent the function $v'(u, v)$ by at most two non-zero probability values. Each three-dimensional array representing P can be represented by a two-dimensional array with indices and values of non-zero elements. This method is often called zero-compression [1]. The analytical complexity of the algorithm drops from $O(|\mathcal{U}| \cdot |\mathcal{V}|^2 \cdot n)$ to $O(|\mathcal{U}| \cdot |\mathcal{V}| \cdot n)$.

[9] We will illustrate the idea for upper restrictions of the speed. An application to lower restrictions of the speed is similar.
[10] We do nothing if it is not the case.

8. Solution using methods of nonlinear optimal control

Optimization theory represents a standard tool to solve these kinds of problems. Optimal control theory is well developed and one can choose from a variety of software tools [15] to find a very-close-to-optimal solution of the problem.

We decided to use ACADO Toolkit [5] – a software environment and an algorithm collection for automatic control and dynamic optimization. We have several reasons for this choice: ACADO is implemented as self-contained code, it is completely written in C++, it is very well documented, it implements single and multiple shooting methods for the numerical solution of optimal control problem, and it is easily configurable. Last but not least, ACADO as a project is very alive, with a strong community eager to help.

The ACADO Toolkit can deal with optimal control problems written in the form of (12). Since ACADO Toolkit authors assume functions to be smooth or at least sufficiently differentiable [6] we approximated (3) using a logistic approximation of the Heaviside step function $H(x)$ (with constant $k = 700$).

Generally, ACADO requires no discretization. But, to be able to incorporate the so-called *path constraints* (different speed limits in different parts of the track), track discretization has to be specified when encoding the problem (12). We chose equidistant track discretizations identical with the one used for the definition of influence diagram to be able to compare the solutions. The discretization of speed and control was not specified.

In this very particular case, ACADO fails in finding a solution without a good initial solution. When we provided an approximate solution found by an influence diagram (even with a very rough discretization) as the initial solution – it speeds up the convergence of ACADO substantially.

9. Experiments

We performed experiments with real data from the bridge version of the Formula 1 circuit in Silverstone. This version of the circuit was used for the British Grand Prix in the years 2000–2009. The Bridge Version is 5.141 km long. In our experiments, we used detailed information from the Silverstone circuit about the GPS coordinates of the path and the speed of four test pilots [23]. The total (lap) time achieved by the best test pilot was 85.51 s. The fastest ever lap time – 78.12 s – was attained by Sebastian Vettel with his Red Bull-Renault when qualifying for the 2009 British Grand Prix [26].

Using a deterministic relation between the variables, we are inevitably working with states of zero probability. If the task is minimize a criterion, the zero probability values may lead to a wrong solution. Therefore, we formulate the problem as a maximization task. Instead of the minimization, we maximize the *savings* with respect to the worst performance. We use the total time savings as the optimality criteria. The time savings at the segment $[i, i+1]$ are defined as

$$f'_{i+1} = f^{max} - f(u_i, v_i), \tag{44}$$

where $f(u_i, v_i)$ is defined by Formulae (19) and (15). The value of f^{max} is the maximum possible time spent in one segment.

Remark 7. For speed values close to zero, the values of $f(u_i, v_i)$ are very high. This would imply high values of f^{max} and might cause rounding errors. To avoid this problem, we disregard speeds lower than 5 m/s in the definition of f^{max}.

We have used various vehicle path, speed and control discretizations during our experiments. We compared solutions given by ACADO with those from Algorithm 1. We experimented with different methods of including speed constraints. We also compared computational demands of different algorithms.

9.1. Comparison of the optimal speed profile with the profile of a test pilot

Fig. 10 contains three speed profiles, the first is computed by ACADO, the second by Algorithm 1 and the third corresponds to a real test pilot performance at the Silverstone F1 circuit (it is obtained using a GPS measurement system [23]). The shaded areas are forbidden by the speed constraints. Note that the testing pilot violates the restrictions several times. Also, the testing pilot's acceleration is slower than expected. The speed constraints used in the model seem to be overly cautious and the car acceleration ability a bit exaggerated. The speed profiles obtained by ACADO and Algorithm 1 overlap each other. To illustrate how much they overlap, their difference is depicted in Fig. 11.

9.2. Comparison of the influence diagram solutions for different discretizations and different methods for speed constraints

Fig. 12 illustrates the behavior of various methods implementing speed constraints (see Sections 7.1, 7.2, and 7.3). To emphasize the difference, we used a relatively rough discretization of the speed and control variables ($|\mathcal{U}| = |\mathcal{V}| = 100$). Discretization of the track was $\Delta s = 5$ m. The ACADO speed and control profiles are used as the solution that the methods are compared with. In the control profiles we can see that the first two methods for including speed constraints (Sections 7.1 and 7.2) suffer by oscillations near the border speed. This problem is almost eliminated by the computational enhancements described in Section 7.3. The differences of the profiles from the one given by ACADO are illustrated in Fig. 13.

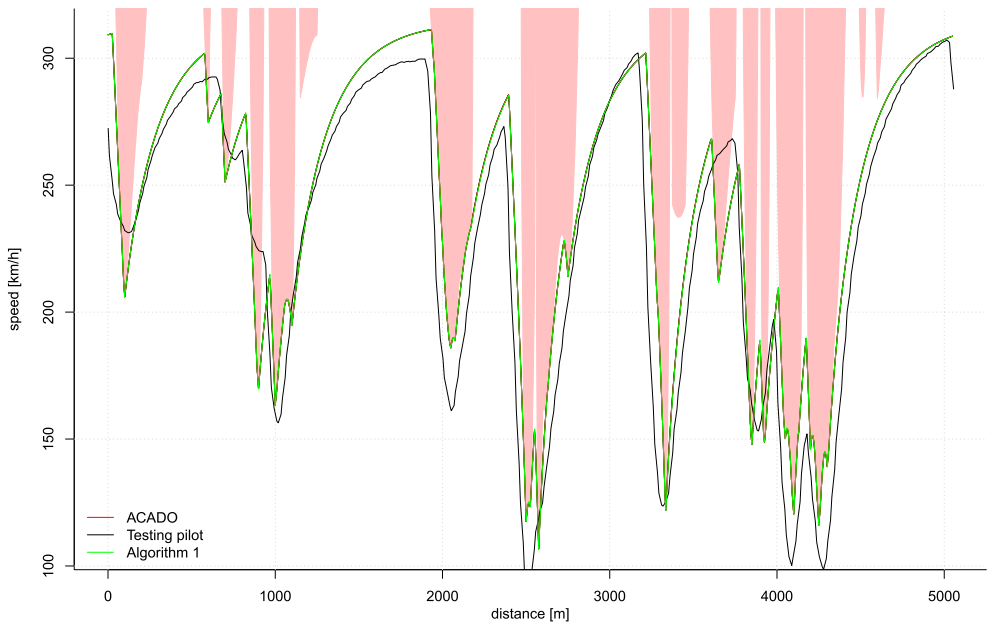

Fig. 10. A comparison of speed profiles obtained by ACADO, Algorithm 1, with a real profile of a testing pilot measured by GPS system [23].

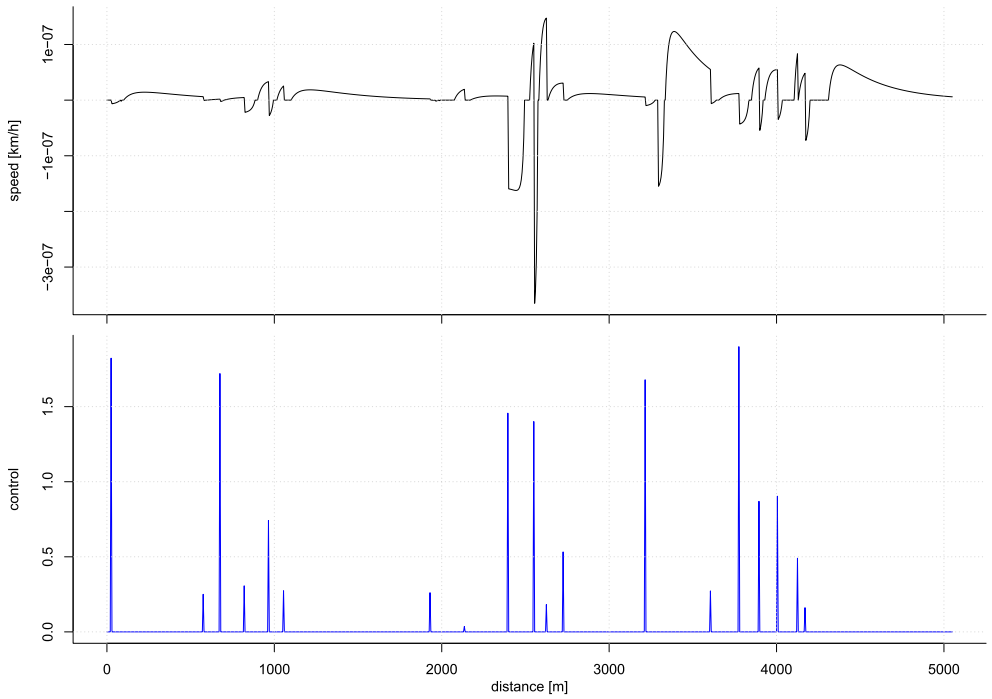

Fig. 11. Differences between speed and control profiles generated by Algorithm 1 and ACADO ($\Delta s = 5$ m).

As mentioned earlier, the occurrence of oscillations of the control when driving with a speed value close to the border one depends on the speed discretization. The finer the speed discretization, the smaller the oscillations and the closer the speed value to the border one.

9.3. Evaluation of the speed profile quality

As a criteria for the quality of the speed profiles we used the Euclidean distance of a given speed profile from the optimal one computed by Algorithm 1. The results are summarized in Table 1 for various speed and control discretizations. Note that $\Delta s = 5$ in this case.

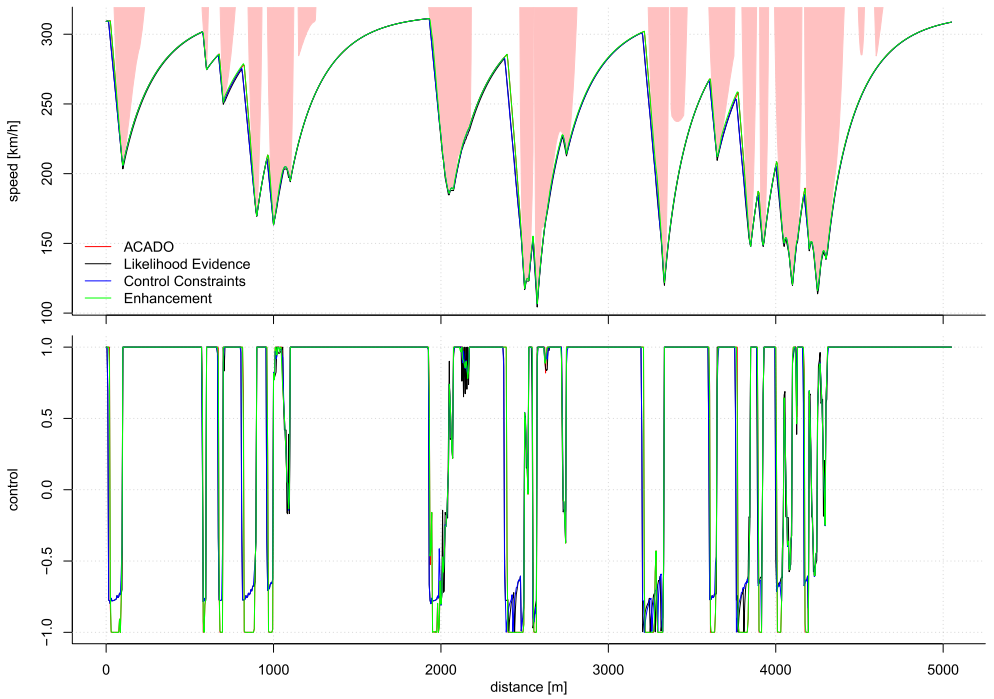

Fig. 12. Speed and control profile for various methods ($|\mathcal{V}| = |\mathcal{U}| = 100$).

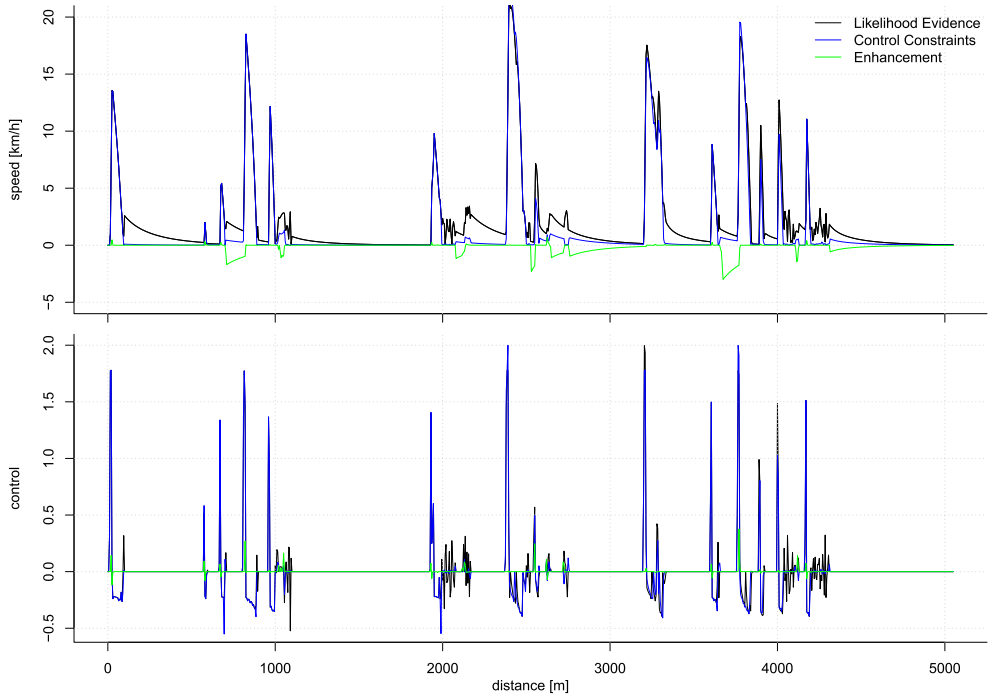

Fig. 13. Differences between optimal profile from Algorithm 1 and profiles generated by various methods ($|\mathcal{V}| = |\mathcal{U}| = 100$).

In Table 2 we compare the estimated total driving time by ACADO and Algorithm 1. The differences are mainly caused by a different track discretization since some speed constraints are omitted and interpolated in the case of a rough discretization step. The most precise estimate should be the one given by Algorithm 1 for the finest discretization.

Table 1
Euclidean distance of speed profiles from the profile of Algorithm 1.

$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{U}\rvert$	Likelihood evidence	Control constr.	Enhancement
25	25	844.794	848.061	97.225
25	50	845.486	846.729	97.263
50	25	234.245	195.261	61.302
50	50	232.035	193.823	61.265
50	100	230.982	193.169	61.298
100	50	136.348	130.015	15.028
100	100	135.295	129.755	15.066
200	100	73.385	68.221	7.249
400	100	30.392	26.531	1.585

Table 2
Estimated total driving time by ACADO and Algorithm 1.

Δs	ACADO	Algorithm 1
20 m	78.17	78.30
10 m	78.52	78.73
5 m	78.69	78.81
1 m	NA	78.84

Table 3
Computational time needed by different methods.

Δs	ID 100×100	ID 100×400	ID 100×800	ACADO	sqp it	qp it
20 m	4.14	16.34	33.36	25.02	9	259
10 m	7.59	25.05	52.17	171.61	9	508
5 m	10.97	43.67	88.87	2049.59	10	1016
1 m	48.61	202.95	434.19	–	–	–

9.4. Computational complexity of the methods used

In the last table – Table 3 – we compare the computational complexity of discrete influence diagram inference methods with ACADO. Various discretizations of $\mathcal{U}$ and $\mathcal{V}$ are denoted as "ID $|\mathcal{U}| \times |\mathcal{V}|$" in the header of Table 3. Results given by ACADO are more precise; however, ACADO needs a good initial solution to be able to solve the problem. This is the case when a solution obtained from an influence diagram is used. In that case, ACADO finishes the task using about 10 iterations of the sequential quadratic programming (denoted as *sqp it* in Table 3), each with a different number of quadratic programming iterations (denoted as *qp it*).

Nevertheless, even so, the time needed to find the solution is much longer than for influence diagrams. For a finer discretization of the path $\Delta s = 1$ ACADO was not even able to complete the task despite a good initial solution. Note that while for influence diagrams the computational time increases linearly (Section 7), with finer track discretization the dependence of ACADO is exponential.

10. Discussion

We proposed an application of influence diagrams to speed profile optimization and tested it in a real-life scenario. We have been able to find optimal solutions efficiently. We verified that these solutions are in accordance with the analytical solution of the considered problem.

In this paper we perform all experiments on an F1 circuit that lies on a flat surface – i.e., we assume the zero inclination angle $\beta(s)$ at all track positions s. Non-zero inclination angles mean that the conditional probability tables $P(V_{i+1}|U_i, V_i)$ describing the vehicle dynamics are different at each path segment i. This does not cause any serious computational problems. We only need to generate these CPTs when they are needed. Actually, in our code used in computational experiments this option is already implemented.

The proposed method allows the application of influence diagrams to more complex scenarios of a speed profile optimization. Speed constraints can be invoked not only by the path radii, but also by other causes like traffic regulations, weather conditions, distance to other vehicles, etc.

Moreover, these conditions can change dynamically. Also, the criteria to be optimized need not be the total time only. We can also consider the safety, fuel consumption, etc. We can do the computations efficiently as long as the utility function is additively decomposed along the path segments.

We also tested our algorithm in a different scenario where the optimization criterion was a weighted average of the total driving time and the total fuel consumption on a road with a varying inclination angle. Numerical optimization methods

can be used to solve this problem. The influence diagrams seem suitable here as well since the optimization criteria are also additively decomposed along the path. Our preliminary experiments (not reported in this paper) suggest that we can get good solutions quickly for these scenarios as well.

We believe that influence diagrams are very appropriate for dynamically changing environments since optimum policies are precomputed for any speed the vehicle can attain. The optimal speed profile can be quickly updated if the conditions change. Influence diagrams are also especially handy in more complex real-life scenarios where the analytic solution is unknown.

Appendix A. Proof of Theorem 1

Proof. For any $j = 1, \ldots, n$ we will denote the joint probability distribution as

$$P(U_1, \ldots, U_j, V_1, \ldots, V_j) = P(V_1) \cdot \prod_{i=2}^{j} P(V_i|U_{i-1}, V_{i-1}) \cdot \delta(U_{i-1}|V_{i-1})$$

and the total utility as

$$f(V_1, \ldots, V_j) = \sum_{i=1}^{j-1} f(V_i, V_{i+1}) \ .$$

For the maximal expected value it holds that

$$\begin{aligned} E_f^* &= \max_{U_1,\ldots,U_{n-1}} \mathop{\mathcal{M}}_{V_1,\ldots,V_n} \Big(P(U_1, \ldots, U_{n-1}, V_1, \ldots, V_n) \cdot f(V_1, \ldots, V_n) \Big) \\ &= \max_{U_1,\ldots,U_{n-1}} \mathop{\mathcal{M}}_{V_1,\ldots,V_n} \left(\begin{array}{l} P(U_1, \ldots, U_{n-1}, V_1, \ldots, V_n) \\ \cdot \Big(f(V_1, \ldots, V_n) + \psi(V_n) \Big) \end{array} \right) \\ &= \max_{U_1,\ldots,U_{n-1}} \mathop{\mathcal{M}}_{V_1,\ldots,V_{n-1}} \left(\begin{array}{l} P(U_1, \ldots, U_{n-1}, V_1, \ldots, V_{n-1}) \\ \\ \cdot \sum_{V_n} P(V_n|V_{n-1}, U_{n-1}) \cdot \left(\begin{array}{l} f(V_1, \ldots, V_{n-1}) \\ + f(V_{n-1}, V_n) \\ + \psi(V_n) \end{array} \right) \end{array} \right) . \end{aligned} \tag{A.1}$$

We can write

$$E_f^* = \max_{U_1,\ldots,U_{n-1}} \mathop{\mathcal{M}}_{V_1,\ldots,V_{n-1}} \left(\begin{array}{l} P(U_1, \ldots, U_{n-1}, V_1, \ldots, V_{n-1}) \\ \cdot \Big(\xi(V_1, \ldots, V_{n-1}) + \psi(U_{n-1}, V_{n-1}) \Big) \end{array} \right) \ ,$$

where

$$\xi(V_1, \ldots, V_{n-1}) = \mathop{\mathcal{M}}_{V_n} \Big(P(V_n|V_{n-1}, U_{n-1}) \cdot f(V_1, \ldots, V_{n-1}) \Big) \tag{A.2}$$

$$\psi(U_{n-1}, V_{n-1}) = \mathop{\mathcal{M}}_{V_n} P(V_n|V_{n-1}, U_{n-1}) \cdot \Big(f(V_{n-1}, V_n) + \psi(V_n) \Big) \ . \tag{A.3}$$

Equation (A.2) can be simplified to

$$\xi(V_1, \ldots, V_{n-1}) = \Big(\mathop{\mathcal{M}}_{V_n} P(V_n|V_{n-1}, U_{n-1}) \Big) \cdot f(V_1, \ldots, V_{n-1}) \tag{A.4}$$

$$= f(V_1, \ldots, V_{n-1}) \ , \tag{A.5}$$

where the second transformation is due to $\mathop{\mathcal{M}}_{V_n} P(V_n|V_{n-1}, U_{n-1}) = 1$. This implies

$$E_f^* = \max_{U_1,\ldots,U_{n-1}} \mathop{\mathcal{M}}_{V_1,\ldots,V_{n-1}} \left(\begin{array}{l} P(U_1, \ldots, U_{n-1}, V_1, \ldots, V_{n-1}) \\ \cdot \Big(f(V_1, \ldots, V_{n-1}) + \psi(U_{n-1}, V_{n-1}) \Big) \end{array} \right) \ .$$

As the next step, we will, for each $v_{n-1} \in \mathcal{V}$, find a value u_{n-1} of decision variable U_{n-1} that maximizes E_f over the terms containing U_{n-1}. Note that the value of U_{n-1} cannot influence the past since the value of V_{n-1} is already known when deciding on U_{n-1}. It means that the values of V_{n-1} effectively separate the influence diagram into two parts and maximization over U_{n-1} can be performed only in the part containing U_{n-1}:

$$E_f^* = \max_{U_1,\ldots,U_{n-2}} \mathop{\mathcal{M}}_{V_1,\ldots,V_{n-1}} \left(\begin{array}{l} P(U_1,\ldots,U_{n-2},V_1,\ldots,V_{n-1}) \\ \cdot \max\limits_{U_{n-1}} \delta(U_{n-1}|V_{n-1}) \cdot \left(\begin{array}{l} f(V_1,\ldots,V_{n-1}) \\ +\psi(U_{n-1},V_{n-1}) \end{array} \right) \end{array} \right) \ .$$

Since $f(V_1,\ldots,V_{n-1})$ does not depend on U_{n-1}, we get

$$E_f^* = \max_{U_1,\ldots,U_{n-2}} \mathop{\mathcal{M}}_{V_1,\ldots,V_{n-1}} \left(\begin{array}{l} P(U_1,\ldots,U_{n-2},V_1,\ldots,V_{n-1}) \\ \cdot (f(V_1,\ldots,V_{n-1}) + \psi(V_{n-1})) \end{array} \right) \tag{A.6}$$

where

$$\psi(V_{n-1}) = \max_{U_{n-1}} \psi(U_{n-1},V_{n-1}) \ .$$

From Formula (A.1) we can get Formula (A.6) by substituting $n-1$ for n. Therefore we can repeat the transformations again and again until $n=2$. In the case $n=2$ Formula (A.6) is reduced to

$$E_f^* = \mathop{\mathcal{M}}_{V_1} P(V_1) \cdot \psi(V_1) \ ,$$

which is Formula (31) of the Theorem we want to prove. □

References

[1] S.K. Andersen, K.G. Olesen, F.V. Jensen, F. Jensen, HUGIN: a shell for building Bayesian belief universes for expert systems, in: G. Shafer, J. Pearl (Eds.), Readings in Uncertain Reasoning, Kaufmann, 1990, pp. 332–337.
[2] R. Bellman, Dynamic Programming, Princeton University Press, 1957.
[3] D.J. Chang, E.K. Morlok, Vehicle speed profiles to minimize work and fuel consumption, J. Transp. Eng. 131 (3) (2005) 173–182.
[4] E. Hellström, J. Åslund, L. Nielsen, Design of an efficient algorithm for fuel-optimal look-ahead control, Control Eng. Pract. 18 (11) (2010) 1318–1327.
[5] B. Houska, H. Ferreau, M. Diehl, ACADO toolkit – an open source framework for automatic control and dynamic optimization, Optim. Control Appl. Methods 32 (3) (2011) 298–312.
[6] B. Houska, H. Ferreau, M. Vukov, R. Quirynen, ACADO Toolkit User's Manual, http://www.acadotoolkit.org, 2009–2013.
[7] R.A. Howard, J.E. Matheson, Influence diagrams, in: R.A. Howard, J.E. Matheson (Eds.), Readings on the Principles and Applications of Decision Analysis, vol. II, Strategic Decisions Group, 1981, pp. 721–762.
[8] F. Jensen, Bayesian Networks and Decision Graphs, Springer-Verlag, 2001.
[9] F. Jensen, F.V. Jensen, S.L. Dittmer, From influence diagrams to junction trees, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1994, pp. 367–373.
[10] D.E. Kirk, Optimal Control Theory: An Introduction, Prentice Hall, Inc., New York, 1971.
[11] V. Kratochvíl, J. Vomlel, Influence diagrams for the optimization of a vehicle speed profile, in: Proceedings of the Twelfth Annual Bayesian Modeling Applications Workshop, in: CEUR Workshop Proceedings, 2015, pp. 44–53, http://ceur-ws.org/Vol-1565/bmaw2015_paper7.pdf.
[12] S.L. Lauritzen, D. Nilsson, Representing and solving decision problems with limited information, Manag. Sci. 47 (9) (2001) 1235–1251.
[13] Y. Li, P.P. Shenoy, A framework for solving hybrid influence diagrams containing deterministic conditional distributions, Decis. Anal. 9 (1) (2012) 55–75.
[14] F. Mensing, R. Trigui, E. Bideaux, Vehicle trajectory optimization for application in eco-driving, in: Vehicle Power and Propulsion Conference (VPPC), IEEE, 2011, pp. 1–6.
[15] H.D. Mittelmann, Decision tree for optimization software (Online), http://plato.asu.edu/guide.html, 2016, accessed 27 March 2016.
[16] V.V. Monastyrsky, I.M. Golownykh, Rapid computation of optimal control for vehicles, Transp. Res., Part B, Methodol. 27 (3) (1993) 219–227, http://www.sciencedirect.com/science/article/pii/0191261593900315.
[17] H.A. Rakha, R.K. Kamalanathsharma, K. Ahn, AERIS: Eco-Vehicle Speed Control at Signalized Intersections Using I2V Communication, Tech. Rep. FHWA-JPO-12-063, Virginia Polytechnic Institute and State University and Virginia Tech Transportation Institute, 2012.
[18] Y. Saboohi, H. Farzaneh, Model for developing an eco-driving strategy of a passenger vehicle based on the least fuel consumption, Appl. Energy 86 (10) (2009) 1925–1932.
[19] R.D. Shachter, Evaluating influence diagrams, Oper. Res. 34 (6) (1986) 871–882.
[20] R.D. Shachter, M.A. Peot, Decision making using probabilistic inference methods, in: Proceedings of the Eighth Annual Conference on Uncertainty in Artificial Intelligence, UAI-92, Morgan Kaufmann, San Mateo, CA, 1992, pp. 276–283.
[21] P.P. Shenoy, Valuation based systems for Bayesian decision analysis, Oper. Res. 40 (1992) 463–484.
[22] P.P. Shenoy, J.C. West, Inference in hybrid Bayesian networks using mixtures of polynomials, Int. J. Approx. Reason. 52 (5) (2011) 641–657.
[23] O.T. Solutions, Rt3000 Inertial and GPS Measurement System, Tech. rep., Oxfordshire, UK, 2002.
[24] E. Velenis, P. Tsiotras, Minimum-time travel for a vehicle with acceleration limits: theoretical analysis and receding-horizon implementation, J. Optim. Theory Appl. 138 (2) (2008) 275–296, http://dx.doi.org/10.1007/s10957-008-9381-7.
[25] J. Vomlel, V. Kratochvíl, Influence diagrams for speed profile optimization: computational issues, in: Proceedings of the Tenth Workshop on Uncertainty Processing, WUPES'15, Nakladatelství Oeconomica, 2015, pp. 203–216, http://wupes.fm.vse.cz/2015/data/Proceedings.pdf.
[26] Wikipedia, 2009 British Grand Prix (Online), https://en.wikipedia.org/wiki/2009_British_Grand_Prix, 2016, accessed 27 March 2016.