

Eco-Driving in Urban Traffic Networks using Traffic Signal Information

Giovanni de Nunzio, Carlos Canudas de Wit, Philippe Moulin, Domenico Di Domenico

▶ **To cite this version:**

Giovanni de Nunzio, Carlos Canudas de Wit, Philippe Moulin, Domenico Di Domenico. Eco-Driving in Urban Traffic Networks using Traffic Signal Information. CDC 2013 - 52nd IEEE Conference on Decision and Control, Dec 2013, Florence, Italy. 10.1109/CDC.2013.6759995 . hal-00909223

HAL Id: hal-00909223

https://inria.hal.science/hal-00909223

Submitted on 26 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Eco-Driving in Urban Traffic Networks using Traffic Signal Information

Giovanni De Nunzio, Carlos Canudas de Wit, Philippe Moulin, Domenico Di Domenico

***Abstract*— This work addresses the problem of finding energy-optimal velocity profiles for a vehicle in an urban traffic network. Assuming communication between infrastructure and vehicles (I2V) and a complete knowledge of the upcoming traffic lights timings, a preliminary velocity pruning algorithm is proposed in order to identify the feasible region a vehicle may travel along in compliance with city speed limits. Then, a graph discretizing approach is utilized for advanced selection, among the feasible "green windows", of the optimal ones in terms of energy consumption. Finally, a velocity trajectory is advised, which will be tracked by the driver-in-the-loop in order to pass through the signalized intersections without stopping. The proposed eco-driving assistance algorithm results are compared to the optimal solution provided by the Dynamic Programming, in order to prove not only the effectiveness but also its capability to be employed online due to its low computational load.**

***Index Terms*— Energy minimization, velocity planning, traffic lights, graph.**

I. INTRODUCTION

The European Union has set binding legislation to cut emissions by a 20% with respect to 1990 levels by the end of 2020 [1]. Traffic congestion and idling time at signalized intersections are among the main causes of energy consumption, and congestion in 2010 has been calculated to cost to drivers in the US about 101 billions of dollars [2].

Recent technology advancements in wireless communication, the diffusion of dedicated protocols [3] for Vehicular Ad-hoc Networks (VANETs), and the availability of low-cost GPS receivers, arose the opportunity to rely on communication between the different agents of an urban traffic network for robust and affordable traffic responsive control strategies. Specifically, Infrastructure-to-Vehicle (I2V) and Vehicle-to-Vehicle (V2V) communications attracted the attention of many due to their potential to enable fast and cheap advanced driving assistance systems (ADAS). Several international projects (Drive C2X, iTetris, COMeSafety2, PATH), involving both automotive manufacturers and research centers, have been started to set up the communication infrastructure and assessing the impact of this technology on traffic management and energy consumption.

Knowledge of the traffic lights signal timings has been proven to have significant benefits on energy efficient use of vehicles (i.e. eco-driving) in urban traffic. Green lights optimal speed advisory (GLOSA) service has been simulated, also varying the penetration rate of the technology, showing promising advantages as compared to ordinary vehicles in terms of fuel consumption and average stop time [8]. Vehicle energy consumption is strictly related to acceleration driving profiles. Soft decelerations to traffic lights on red, using advanced wireless transmitted information, have been proven to be more energy efficient than sudden stops, allowing a reduction in both fuel consumption and emissions [6]. Other works show how it is possible to reduce energy consumption by preventing the vehicle from coming to a full stop at the intersections and by advising cruising velocities in order to catch as many traffic lights on green as possible. In [7], authors developed an algorithm to minimize the acceleration profile while driving through a sequence of signalized intersections, simulating scenarios with stochastically varied parameters and showing a systematic reduction in fuel consumption, emissions and travel time. Interesting results on eco-driving with probabilistic signal phase and timing (SPAT) information are shown in [9], where a comparison between an uninformed driver and a driver with different information levels (full horizon and short-term) is conducted using Dynamic Programming (DP). An innovative solution to the velocity advisory problem, in proximity to a traffic light, was proposed in [10], where smartphones equipped with cameras mounted on the windshield of the vehicles allow to build a SPAT map of every intersection by relying only on V2V communication, without directly interacting with the infrastructure. This approach enables GLOSA service and adaptive route change (ARC) among the others. Predictive cruise control (PCC) in traffic networks with signalized intersections was extensively treated in [11], where a preliminary logic computes the constant velocity to be tracked by the driver to pass through the as high number of intersections as possible, while penalizing the brakes action to indirectly reduce energy consumption and trip time. In [14] only one traffic light is considered and accelerations upstream and downstream of the intersection are varied to find sub-optimal

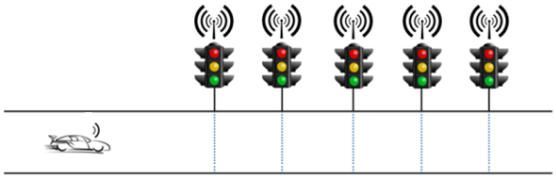


Fig. 1: I2V communication in an urban traffic network.

G. De Nunzio, Ph.D. Candidate at GIPSA-Lab and IFPen, France, *giovanni.de-nunzio@ifpen.fr*

C. Canudas de Wit, Director of Research at CNRS, GIPSA-Lab, NeCS team, Grenoble, France, *carlos.canudas-de-wit@gipsa-lab.fr*

P. Moulin, Department of Control Signal and Systems at IFPen, Rueil-Malmaison, France, *philippe.moulin@ifpen.fr*

D. Di Domenico, Department of Control Signal and Systems at IFPen, Solaize, France, *domenico.didomenico@ifpen.fr*

values for the fuel consumption minimization problem. In [15] Dynamic Programming is used to find the minimum energy solution, for a trip with up to two traffic lights and investigating different timings of the green signal.

In the case of long or multiple available "green windows" at each intersection, different driving profiles are possible and an optimal route decision should be made at the beginning of the driving horizon to keep the energy consumption at a minimum. The novelty of the proposed approach lies in the capability of identifying in advance a low cost path (eco-path), among all the possible trajectories within the driving horizon, without having to stop at traffic lights, without penalizing the trip time and while keeping a low computational load. The described algorithm provides a fast sub-optimal solution for the energy minimization problem, which is promising for an online speed advisory service to the driver, and it is compared to the optimal solution provided offline by Dynamic Programming (benchmark). Note that unexpected additional constraints coming from the network (vehicles, pedestrians, etc.) may be addressed with new executions of the algorithm.

Section II describes the problem and its elements from a mathematical point of view. In Section III the applied methodology for the solution of the problem is explained in detail. Section IV contains simulation results and comparisons of the algorithm against the benchmark. Conclusions and future developments in Section V.

II. PROBLEM FORMULATION

From a mathematical point of view, a constrained optimization problem needs to be solved. The function to be minimized is the energy consumption over the trip horizon, subject to the vehicle dynamic model, with constraints on the crossing times of the intersections, allowed only when the relative traffic light state is on "green".

A. Vehicle model

Vehicles in our scenario are assumed to be electric. The equivalent circuit model of a DC motor gives [5]:

$$\begin{aligned} V_a &= i_a R_a + e \\ e &= \kappa \omega_m \\ T_m &= \kappa i_a \end{aligned} \tag{1}$$

where V_a is the armature voltage, i_a is the armature current, R_a is the armature resistance, e is the back electromotive force, κ is the speed constant, ω_m is the rotational speed of the motor, T_m is the motor torque. The electric power is calculated by expressing:

$$P_m = V_a i_a = \omega_m T_m + \frac{R_a}{\kappa^2} T_m^2 \tag{2}$$

In this equation $V_a i_a$, that is the electric power supplied to the armature, is converted into actual mechanical power developed by the armature ($\omega_m T_m$), and electric power wasted in the armature ($\frac{R_a}{\kappa^2} T_m^2$). The control variable u is the demanded motor torque:

$$T_m = u \tag{3}$$

The longitudinal dynamic model is:

$$m \frac{dv}{dt} = F_t - F_{\text{aero}} - F_{\text{friction}} - F_{\text{slope}} \tag{4}$$

where F_t is the traction force at the wheels, F_{aero} is the aerodynamic force, F_{friction} is the rolling resistance force, F_{slope} is the gravity force.

Under the hypotheses of no losses in the transmission and no slip at wheels, the vehicle model shall be written as:

$$\begin{cases} \dot{x} = v \\ m\dot{v} = \frac{uR_t}{r} - \frac{1}{2}\rho_a A_f c_d v^2 - mgc_r - mg\sin(\alpha(x)) \end{cases} \tag{5}$$

where R_t is the transmission ratio, r is the wheel radius, m is the vehicle mass, ρ_a is the external air density, A_f is the vehicle frontal area, c_d is the aerodynamic drag coefficient, c_r is the rolling resistance coefficient, α is the road slope as a function of the position x, and g is the gravity.

The sum of the aerodynamic and rolling frictions can be approximated as a second order polynomial:

$$F_{\text{res}} = F_{\text{aero}} + F_{\text{friction}} = a_0 + a_1 v + a_2 v^2 \tag{6}$$

where a_0, a_1 and a_2 are identified parameters [4].

Under these assumptions the vehicle model can be simplified as follows:

$$\begin{cases} \dot{x} = v \\ \dot{v} = h_1 u - h_2 v^2 - h_3 v - h_0 \end{cases} \tag{7}$$

with

$$h_1 = \frac{R_t}{mr}, \quad h_2 = \frac{a_2}{m}, \quad h_3 = \frac{a_1}{m}, \quad h_0 = \frac{a_0}{m} + g\sin(\alpha) \tag{8}$$

Then the power consumption can be expressed as:

$$P = b_1 uv + b_2 u^2 \tag{9}$$

where

$$b_1 = \frac{R_t}{r}, \quad b_2 = \frac{R_a}{\kappa^2} \tag{10}$$

B. Traffic lights timing

The control of the intersection signal timings is not considered here, therefore cycle, stage and offset times are deterministic and given. Under the assumption that traffic lights can share with other traffic agents their signal timings information (I2V communication), it is possible to formulate mathematically the state evolution of the traffic lights as follows:

$$s_i(t) = \begin{cases} 1, & (k-1)T < t - l_i \leqslant (k-1)T + T_{gr} \\ 0, & (k-1)T + T_{gr} < t - l_i \leqslant kT \end{cases} \tag{11}$$

$$l_i = \begin{cases} T_0 + \frac{d_i}{\bar{v}}, & T_0 + \frac{d_i}{\bar{v}} < T \\ T_0 + \frac{d_i}{\bar{v}} - kT, & kT < T_0 + \frac{d_i}{\bar{v}} < (k+1)T \end{cases} \tag{12}$$

where s_i is the state of the i-th traffic light, $T_0 \in [0, T]$ is the initial time to be chosen for the cycles, k is the number of cycles, T is the time duration of a cycle, T_{gr} is the duration of the green signal, d_i is the position of the i-th traffic light, $\bar{v}$ is the "green wave" velocity, $k \in \mathbb{Z}$, $i \in [1, \ldots, n]$, being n the number of intersections. This is explained as follows:

- the time $(d_i/\bar{v})$ to reach each intersection traveling at the velocity $\bar{v}$ is calculated;
- T_0, chosen randomly in the interval $[0, T]$, is added to the "green wave" condition just calculated in order to generate a random starting point in the signal cycle;
- hence we obtain the initial "time offset" (l_i) of the lights, which is scaled down to the interval $[0, T]$ if it exceeds the duration T of the cycle;
- as the simulation time t runs, the state of the lights is updated according to the cycle and split times.

C. Optimal control problem

Finally, the optimization problem may be stated as follows:

$$\begin{cases} \min\limits_{u} & J = \int\limits_{t_0}^{t_f} b_1 uv + b_2 u^2 \, \mathrm{d}t \\ \text{s.t.} & \dot{x} = v \\ & \dot{v} = h_1 u - h_2 v^2 - h_3 v - h_0 \\ & x(t_0) = x_0, \quad x(t_f) = D \\ & x(t_i) = d_i \wedge s_i(t_i) = 1 \\ & v(t_0) = v_0, \quad v(t_f) = v_f \\ & v_{\min} \leq v \leq v_{\max} \\ & u_{\min} \leq u \leq u_{\max} \end{cases} \tag{13}$$

Nature and complexity of the constraints (14)

$$x(t_i) = d_i \wedge s_i(t_i) = 1 \tag{14}$$

which may result in disjoint sets due to the presence of multiple available "green windows" at the same intersection, affect the convexity of the problem and a global analytical solution is not achievable. In other words the constraints (14) represent a non-convex set and the nonlinear objective function, because of such a discontinuity in the constraints, assumes multiple local minima. Therefore, the solution to this optimal control problem has to be sought in a sub-optimal way, making use of a set of algorithms to simplify the control envelop, recover the convexity of the constraints set and advise the driver on the velocity to track.

III. OPTIMIZATION ALGORITHM

The original optimization problem (13) will be now divided into sub-problems for the simplification of constraints (14) and the formulation of a convex optimization problem.

The methodology may be summarized in three main steps:

1) Pruning algorithm to identify the set of feasible "green windows" and their portions which allow not to stop at the intersections in accordance with city speed limits $(v \in [v_{\min}, v_{\max}])$;
2) Construction of a weighted directed acyclic graph (nodes at each feasible "green window") to analyze the energy cost of all the possible paths in the driving horizon, and identification, by means of Dijkstra's algorithm, of the "shortest" (in terms of energy) path;
3) Solution of an optimization problem to obtain the optimal crossing points at each intersection through the previously selected "green windows".

A. Pruning algorithm

The velocity pruning algorithm reduces the number of available "green windows" by identifying only the feasible ones that allow to reach a horizon at a specified final time without stopping. Let t_i be the crossing time at the i-th intersection, (t_0, d_0) the initial position of the vehicle, d_i the position of the i-th traffic light, D the horizon.

Algorithm 1 Pruning algorithm

for $i \leftarrow 1$ to n **do**
  $t_{i,\min} = \frac{d_i - d_{i-1}}{v_{\max}} + t_{i-1,\min}$;
  $t_{i,\max} = \frac{d_i - d_{i-1}}{v_{\min}} + t_{i-1,\max}$;
  if $t_{i,\min} \notin$ GREEN **then**
    $t_{i,\min} =$ first instant of next GREEN;
  end if
  if $t_{i,\max} \notin$ GREEN **then**
    $t_{i,\max} =$ last instant of previous GREEN;
  end if
  $t_{i,\max} = min\left\{t_{i,\max}, \frac{D - d_i}{v_{\max}}\right\}$;
end for
for $i \leftarrow n$ to 1 **do**
  if $t_{i,\max} \leq t_{i-1,\max} \;||\; \frac{d_i - d_{i-1}}{t_{i,\max} - t_{i-1,\max}} > v_{\max}$ **then**
    $t_{i-1,\max} = t_{i,\max} - \frac{d_i - d_{i-1}}{v_{\max}}$;
  end if
end for

The algorithm will output $[t_{i,\min}, t_{i,\max}]$, which represent respectively the minimum and the maximum feasible crossing times at the i-th intersection, respecting speed limits, and its steps may be described as follows:

1) from the starting point calculate $[t_{\min}, t_{\max}]$ corresponding to the $[v_{\min}, v_{\max}]$ to reach the first intersection d_1;
2) check if from the point with coordinates $(t_{\max}, d_1)$ it is possible to reach the final destination in compliance with the $v_{\max}$ constraint, if not set $t_{\max}$ accordingly;
3) from d_1 use the two starting points $(t_{\min}, d_1)$ and $(t_{\max}, d_1)$ to calculate $t_{\min}$ and $t_{\max}$ at d_2;
4) check again if from the point $(t_{\max}, d_2)$ it is possible to reach the final destination in compliance with the $v_{\max}$ constraint, if not set $t_{\max}$ accordingly;
5) repeat steps 3 and 4 for each intersection up to d_n;
6) at the last intersection d_n connect the two points $(t_{\min}, d_n)$ and $(t_{\max}, d_n)$ to the destination;
7) check backwards from d_n to d_1 if any $(t_{\max}, d_i) \leq (t_{\max}, d_{i-1})$ or if any $t_{\max}$ induces higher velocities than $v_{\max}$, if so move $(t_{\max}, d_{i-1})$ accordingly.

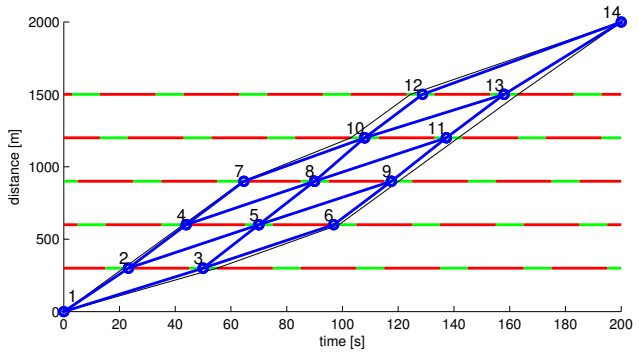


Fig. 2: Graph of the possible paths within the feasibility region, with 1 node per "green window" and node numbering.

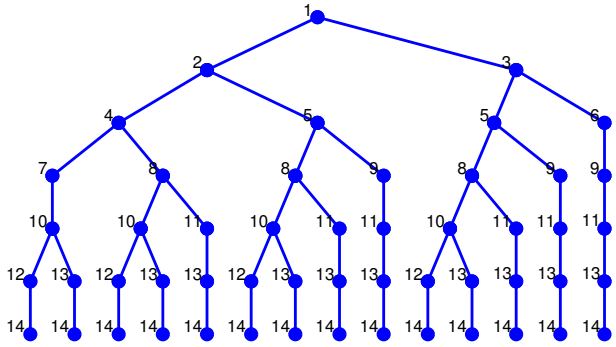


Fig. 3: Tree expansion of the possible paths from the root to the destination node.

In case no feasible region is found, it will not be possible to avoid stopping at a traffic light along the trip, but the algorithm may be run again at the next "green" signal with respect to the moving horizon.

B. Optimal path

After running Algorithm 1, many possible paths are still present and a decision for the optimal one needs to be made.

The idea is to approximate, by means of a weighted directed acyclic graph (DAG), the energy cost of all the possible paths from the initial to the final point within the feasibility region shown in Fig.2.

Therefore, let $\mathcal{G} = (V, E)$ be such a graph, where V is the set of vertices representing the crossing times at each feasible "green window", and E is the set of feasible paths connecting the vertices of the graph. Let us also define a weighting function $w : E \to W$ which associates each edge of the graph with a weight. In this application the weight W is the energy cost to travel along the path and it may be seen as the sum of the energy contribution given by a constant speed trip on an edge, and the contribution given by the change of velocity at a node between two edges.

$$W = E_{\text{total}} = E_{\text{link}} + E_{\text{jump}} \tag{15}$$

$$E_{\text{link}} = \int_{t}^{t+\Delta t} b_1 \bar{u}\bar{v} + b_2 \bar{u}^2 \, \mathrm{d}t = \Delta t \, \left(b_1 \bar{u}\bar{v} + b_2 \bar{u}^2 \right) \tag{16}$$

where $\bar{v}$ is a constant traveling velocity on an edge, and from (7), for $\dot{v} = 0$, $\bar{u} = \frac{1}{h_1}\left(h_2 \bar{v}^2 + h_3 \bar{v} + h_0\right)$;

$$E_{\text{jump}} = \int_{0}^{t_{\text{jump}}} b_1 uv + b_2 u^2 \, \mathrm{d}t \tag{17}$$

where $t_{\text{jump}} = \frac{|v_{i+1} - v_i|}{a}$, a is a constant fixed acceleration linearly modeling the velocity in every transient, $v = v_i \pm a \cdot t$ for $i = 0, \ldots, n+1$, with v_0 being the initial velocity and $v_{n+2} = v_f$ the final velocity.

Such graph approximation introduces a difficulty when it comes to assigning the velocity jump energy contribution to the edges. The transient energy contribution on an edge depends on the edge previously traveled: if coming from a higher speed edge the transient contribution will be very small, on the other hand if coming from a lower speed edge the transient contribution will be much bigger due to the increase of velocity. Every node of the graph with more than one incoming edge is critical and needs to be "decoupled" in order to have a correct weight assignment, since the outgoing edges of the critical nodes can have only one weight. The decoupling of the graph into the tree of all the possible paths is shown in Fig.3. It was necessary to create duplicates of the critical nodes in order to have independent paths, and in the case under analysis the initial setting with 14 nodes (Fig.2) results in a bigger network (Fig.3) with the same amount of possible paths from node 1 to node 14 but with 52 nodes.

Well known theorems in graph theory allow to analyze a priori the complexity of this procedure (see p.165 in [13]). Specifically, being A the adjacency matrix of a graph $\mathcal{G} = (V, E)$, the number of paths of length r from a node a to a node b is equal to $(A^r)_{a,b}$.

The number of nodes of the expanded decoupled tree is given by the sum of the entries of the first row of the geometric series of the adjacency matrix up to the length of the trip r. The geometric series generated by the matrix A, if $|\lambda_i| \leq 1$ for each eigenvalue $|\lambda_i|$ of A, is known to converge to

$$\sum_{k=0}^{r} A^k = (I - A)^{-1}(I - A^{r+1}) \tag{18}$$

The problem has an exponential complexity and its computational load might increase significantly but, on a realistic horizon of few intersections with not numerous "green windows" in the feasible region, the number of nodes of the decoupled tree can be kept low.

To transform the decoupled tree in a graph, a fictitious destination node connected to all the leaves is added. On the expanded weighted DAG, Dijkstra's algorithm is used to find the minimum energy trip from the origin to the destination.

C. Simplified optimization problem

The aim of the described simplification steps is to have a convex, also in the constraints, optimization problem to find the optimal crossing times within the "green windows"

selected by Dijkstra's algorithm. The formulation of this optimization problem may be carried out in alternative ways.

1) Velocity as variable: In order to formulate the optimization problem using velocity as variable, let us define the vector of optimization variables (velocities to track in each segment), assuming n to be the number of intersections:

$$\mathbf{X} = [v_1, v_2, \dots, v_n, v_{n+1}]^T \in \mathbb{R}^{n+1} \tag{19}$$

Then a vector of time intervals for each segment is defined:

$$\begin{aligned}\mathbf{T} = \mathbf{T}(\mathbf{X}) &= [t_1 - t_0, t_2 - t_1, \dots, t_n - t_{n-1}, t_f - t_n]^T \\ &= \left[\frac{d_1 - d_0}{v_1}, \frac{d_2 - d_1}{v_2}, \dots, \frac{d_n - d_{n-1}}{v_n}, \frac{D - d_n}{v_{n+1}}\right]^T\end{aligned} \tag{20}$$

where d_i is the known position of the i-th traffic light, D is the destination/horizon, and $\mathbf{T} \in \mathbb{R}^{n+1}$.

The objective function is the same as the one in the original optimization problem (13):

$$J = \int_{t_0}^{t_f} P \,\mathrm{d}t = \int_{t_0}^{t_f} b_1 uv + b_2 u^2 \,\mathrm{d}t \tag{21}$$

By using the same procedure as in the graph weights assignment, we may split the objective function in an energy term related to the contributions of the segments traveled at constant velocity, and energy terms related to the velocity variations between subsequent segments.
Therefore, the objective function may be rewritten as:

$$J(\mathbf{X}) = \mathbf{T}^T \bar{P}(\mathbf{X}) + \sum_{i=0}^{n+1} E_{\text{jump}} \tag{22}$$

where $\bar{P}(\mathbf{X})$ is the power required by the vehicle when traveling at constant velocity (no velocity variation in time):

$$\bar{P}(\mathbf{X}) = b_1 \bar{u}(\mathbf{X})\mathbf{X} + b_2 \bar{u}(\mathbf{X})^2 \tag{23}$$

with

$$\bar{u}(\mathbf{X}) = \frac{1}{h_1}\left(h_2 \mathbf{X}^2 + h_3 \mathbf{X} + h_0\right) \tag{24}$$

Finally the optimization problem may be formulated as:

$$\begin{cases} \min\limits_{\mathbf{X}} \; J(\mathbf{X}) \\ v_{\min} \le v_i^-(\mathbf{X}) \le v_i \le v_i^+(\mathbf{X}) \le v_{\max} \end{cases} \tag{25}$$

Note that in this case, the constraints are nonlinear. Specifically the bounds of the constant velocity to be tracked in each segment (v_i) depend on the crossing time at the previous intersection (t_{i-1}) and experimentally they are found to be approximated by cubic functions.

2) Time as variable: In order to overcome the difficulty represented by the nonlinear constraints of the previous formulation, it is possible to formalize the problem in an analogous way which allows to deal with constant constraints. Let us first define a vector of optimization variables (crossing times at intersections), supposing n to be the number of intersections:

$$\mathbf{X} = [t_1, t_2, \dots, t_n]^T \in \mathbb{R}^n \tag{26}$$

Then a vector of time intervals for each segment is defined:

$$\mathbf{T} = \mathbf{T}(\mathbf{X}) = [t_1 - t_0, t_2 - t_1, \dots, t_n - t_{n-1}, t_f - t_n]^T \tag{27}$$

Knowing the position of the traffic lights, let us then define the vector of the constant velocities in each segment:

$$\begin{aligned}\mathbf{V} = \mathbf{V}(\mathbf{X}) &= [v_1, v_2, \dots, v_n, v_{n+1}]^T \\ &= \left[\frac{d_1 - d_0}{t_1 - t_0}, \frac{d_2 - d_1}{t_2 - t_1}, \dots, \frac{d_n - d_{n-1}}{t_n - t_{n-1}}, \frac{D - d_n}{t_f - t_n}\right]^T\end{aligned} \tag{28}$$

where d_i is the position of the i-th traffic light with respect to the origin, D is the destination, t_f is the horizon time, and $\mathbf{T} \in \mathbb{R}^{n+1}$, $\mathbf{V} \in \mathbb{R}^{n+1}$.

The objective function is again the same as the one in the original optimization problem (13):

$$J = \int_{t_0}^{t_f} P \,\mathrm{d}t = \int_{t_0}^{t_f} b_1 uv + b_2 u^2 \,\mathrm{d}t \tag{29}$$

It may be rewritten as:

$$J(\mathbf{X}) = \mathbf{T}^T \bar{P}(\mathbf{V}) + \sum_{i=0}^{n+1} E_{\text{jump}} \tag{30}$$

where $\bar{P}(\mathbf{V})$ is the power required by the vehicle when traveling at constant velocity (no velocity variation in time):

$$\bar{P}(\mathbf{V}) = b_1 \bar{u}(\mathbf{V})\mathbf{V} + b_2 \bar{u}(\mathbf{V})^2 \tag{31}$$

with

$$\bar{u}(\mathbf{V}) = \frac{1}{h_1}\left(h_2 \mathbf{V}^2 + h_3 \mathbf{V} + h_0\right) \tag{32}$$

Finally the optimization problem may be formulated as:

$$\begin{cases} \min\limits_{\mathbf{X}} \; J(\mathbf{X}) \\ t_i^- \le t_i \le t_i^+ \end{cases} \tag{33}$$

where t_i^- and t_i^+ are constants and represent the ends of the selected green split time at each intersection.

IV. SIMULATION RESULTS

In the following, performance of the proposed algorithm is compared to the solution provided by DP [12].

TABLE II: Simulation parameters

Parameter	Description	Value	Unit
D	horizon	2000	m
t_f	final time	200	s
v_f	final velocity	10	m/s
$v_{\min}$	minimum speed limit	5	m/s
$v_{\max}$	maximum speed limit	14	m/s
m	vehicle mass	1190	kg
r	wheel radius	0.2848	m
R_t	transmission ratio	6.066	

TABLE I: Optimal energy consumption on the paths calculated with Dynamic Programming

Path number	Optimal windows	$v_0 = 10$ m/s	$v_0 = 11$ m/s	$v_0 = 12$ m/s	$v_0 = 13$ m/s	$v_0 = 14$ m/s
1	1-2-4-8-10-12-14	459900 J	446800 J	429200 J	410700 J	394200 J
2	1-2-4-8-10-13-14	436700 J	423400 J	406000 J	387800 J	371500 J
3	1-2-5-8-10-12-14	454700 J	438200 J	421600 J	406700 J	397400 J
4	1-3-5-8-10-12-14	450700 J	449700 J	449100 J	449000 J	449100 J
5	1-3-5-8-10-13-14	435800 J	434900 J	434600 J	434800 J	435400 J
6	1-3-5-8-11-13-14	417800 J	418100 J	417700 J	417400 J	418000 J
7	1-3-5-9-11-13-14	420400 J	420000 J	419800 J	419600 J	420400 J
8	1-2-5-9-11-13-14	425100 J	408600 J	392200 J	376900 J	367500 J
9	1-2-4-8-11-13-14	423600 J	410400 J	393200 J	375200 J	359100 J
10	1-2-4-7-10-13-14	427300 J	412700 J	394000 J	378500 J	362300 J
11	1-2-5-8-11-13-14	415100 J	398700 J	382100 J	367300 J	357800 J
12	1-2-4-7-10-12-14	438600 J	423800 J	404400 J	389100 J	373000 J
13	1-3-6-9-11-13-14	413600 J	411800 J	411700 J	411700 J	411800 J
14	1-2-5-8-10-13-14	438300 J	421700 J	405100 J	390200 J	380900 J

A. *Control scenario*

The case under study presents a simplified scenario with one vehicle in the traffic network, traveling through $n = 5$ perpendicular signalized intersections, along a road section of 2000 m on a total time horizon of 200 s. The vehicle's objective is to follow an energy optimal velocity profile which allows to find all the traffic lights on green. The traffic lights obey a known signal timing policy, as specified in (12), and their states are known to the vehicle. The green split time is 10 s, the total cycle time is set to 30 s. Note that, although traffic lights obey a "'green wave" condition, the distribution of the green windows may be such that the velocity of the wave is not energy optimal.

The initial velocity of the vehicle is varied throughout the simulations to assess its impact on the energy consumption and on the optimal path. Traffic lights timings, final velocity, horizon are fixed for consistency in results comparison. The objective of the performed simulations is to prove the importance of a methodology which allows to achieve a fast sub-optimal solution suitable for online implementation and eco-driving assistance service.

B. *Optimal path*

The first part of the presented simulation results aims at validating the pruning algorithm and the optimal path identification. DP, used in the following as a benchmark, provides the optimal solution to the original problem and it has been used to compute the energy cost of all the possible paths in the feasible region.

Simulations for this first validation part have been run varying the initial velocity of the trip, which has an impact on the optimal path. In the scenario under study the vehicle does not stop at any intersection in the feasibility region (Fig.2), nevertheless the energy consumption varies depending on the path choice; in the worst case, the choice of a path rather than the optimal one, results to be 10% more expensive in terms of energy.

Table I summarizes the trip energy consumption of all the possible paths for different choices of initial velocity, and the minimum cost path, for each case, is highlighted in red. The optimal windows numbering refers to the one utilized in Fig.2. In Fig.4, the corresponding column for $v_0 = 10$ m/s of Table I is represented, with the minimum energy path in green and the unconstrained path in blue (no traffic lights, constant velocity trip along the whole horizon).

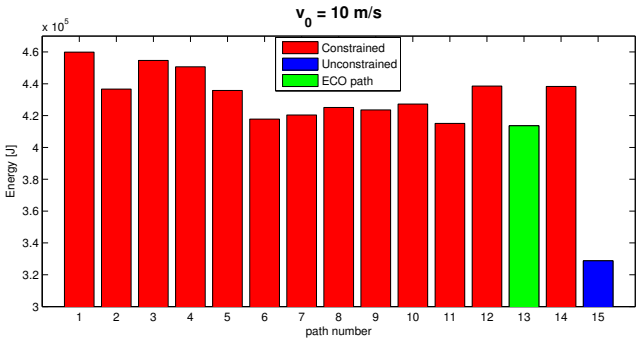


Fig. 4: Optimal energy cost of all the possible paths from initial to final point, for $v_0 = 10$ m/s.

The accuracy of the graph approximation, as well as the computational load, increases with the number of nodes for each "green window". A graph with one node per window, as shown in Fig.2, and a graph with three nodes per window (at the mid-point and at the two ends) have been tested, and the optimal path suggestion obtained with Dijkstra's algorithm is validated against the true minimum energy path, whose energy was computed with the DP.

As summarized in Table III, the graph with 3 nodes per window approximates better the whole energy cost and it fails in two cases, whereas the smaller graph fails in five cases. Note that in this scenario, the algorithm gives correct estimations also for lower initial velocities $v_0 \in [5, 9]$, but it was not reported in the table for brevity. This is a sub-optimal solution, the path identification is not flawless, but in the worst case, for the graph with 3 nodes/window, the algorithm leads to the selection of a path which is only 2% more expensive than the optimal one.

C. *Optimal crossing times*

Once the sub-optimal windows are obtained with Dijkstra's algorithm, the optimal crossing times at each intersec-

TABLE III: Optimal path identification (Graph vs. DP)

Initial velocity	Graph 1 node/window	Graph 3 nodes/window	Optimal
10 m/s	path #13	path #13	path #13
11 m/s	path #13	path #11	path #11
12 m/s	path #10	path #11	path #11
13 m/s	path #10	path #9	path #11
14 m/s	path #10	path #10	path #11

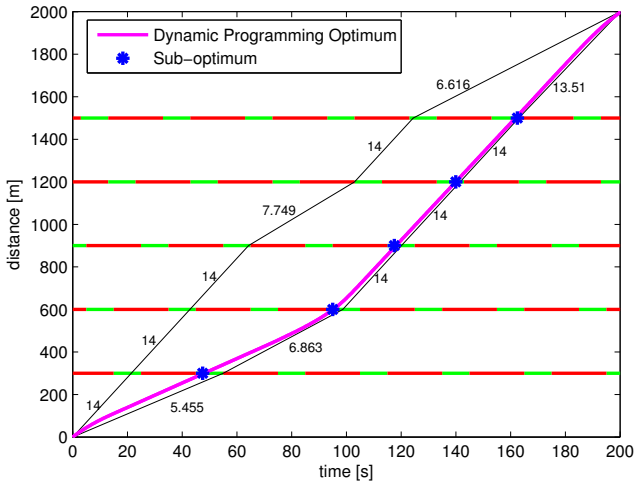


Fig. 5: Sub-optimal crossing times compared to the optimal solution provided by DP, for $v_0 = 10$ m/s.

tion in the horizon need to be calculated in order to provide the driving assistance by suggesting a velocity profile to the driver. The solver employed to solve the optimization problems (25) and (33) in MATLAB® was `fmincon`, using the interior-point method algorithm. As already mentioned, the optimization problem (25) has nonlinear constraints which need to be redefined at each optimization run. Therefore the solution of the optimization (33), using crossing times as variables, benefits from an easier definition and requires less iterations to converge to the solution.

In Fig.5, the optimal solution provided by DP through the optimal windows (path #13) for $v_0 = 10$ m/s is compared to the sub-optimal solution. In Table IV the time deviations of the sub-optimal solution from the DP solution at the five intersections are summarized, both for problem (25) and (33).

The simulations were run with a laptop equipped with an Intel(R) Core(TM) i3-2350M at 2.30GHz and 6GB of RAM. DP takes about 3 hours to run a complete simulation and provide the optimal solution. The proposed methodology takes 4.7 seconds (graph 3 nodes/window) or 2.5 seconds (graph 1 node/window) to run all the presented steps and provide the sub-optimal crossing times at each intersection.

V. CONCLUSIONS

Previous works proved how the exchange of information between the infrastructure and the vehicles in the urban traffic (I2V) has a positive impact on the energy consumption. In this work, the possibility of further improvement

TABLE IV: Time deviation from the optimal crossing times

	Int. #1	**Int. #2**	**Int. #3**	**Int. #4**	**Int. #5**
Opt. TIME	0.4 s	0.03 s	0.5 s	0.16 s	0.8 s
Opt. VELOCITY	0.4 s	0.03 s	0.5 s	0.16 s	0.8 s
Average deviation: 0.38 s					

of energy consumption in case of multiple available no-stop paths on a driving horizon is presented. The proposed methodology also allows to simplify a complex constrained optimization problem, and it becomes appealing for online applications. Indeed the proposed algorithm leads to a sub-optimal solution and, as shown in the results, may identify wrong optimal paths. However, it was shown that the sub-optimal solution is still very close to the optimal one and in terms of energy the difference is minor. Future works will consider also traffic lights with time-varying cycle and stage times, and vehicles embedded in a traffic flow.

REFERENCES

[1] European Commission Climate Action, "What is the EU doing about climate change?", [ONLINE]. Available: `http://ec.europa.eu/clima/policies/brief/eu/index_en.htm`

[2] D. Schrank, T. Lomax and B. Eisele, "2011 Urban Mobility Report", Texas Transportation Institute, [ONLINE]. Available: `http://mobility.tamu.edu/ums/report/`

[3] IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments; available from IEEE Standards.

[4] W. Dib, A. Chasse, A. Sciarretta, P. Moulin, "Optimal Energy Management Compliant with Online Requirements for an Electric Vehicle in Eco-Driving Applications", 2012 IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling.

[5] N. Petit and A. Sciarretta, "Optimal drive of electric vehicles using an inversion-based trajectory generation approach", IFAC World Congress, 2011.

[6] M. Li, K. Boriboonsomsin, G. Wu, W. Zhang, M. Barth, "Traffic Energy and Emission Reductions at Signalized Intersections: A Study of the Benefits of Advanced Driver Information", International Journal of Intelligent Transportation Systems Research, 2009, vol. 7(1), 49-58.

[7] S. Mandava, K. Boriboonsomsin, M. Barth, "Arterial velocity planning based on traffic signal information under light traffic conditions", Proc. 12th International IEEE Conference on Intelligent Transportation Systems ITSC '09, 2009, 1-6.

[8] K. Katsaros, R. Kernchen, M. Dianati, D. Rieck, "Performance study of a Green Light Optimized Speed Advisory (GLOSA) application using an integrated cooperative ITS simulation platform" Proceedings of 7th International Wireless Communications and Mobile Computing Conference (IWCMC), 2011.

[9] G. Mahler, A. Vahidi, "Reducing idling at red lights based on probabilistic prediction of traffic signal timings", Proceedings American Control Conference (ACC), 2012, 6557-6562.

[10] E. Koukoumidis, M. Martonosi, L.S. Peh, "Leveraging Smartphone Cameras for Collaborative Road Advisories", IEEE Transactions on Mobile Computing, 2012, 11, 707-723.

[11] B. Asadi, A. Vahidi, "Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time", IEEE Transactions on Control Systems Technology, 2011.

[12] O. Sundstrom, L. Guzzella, "A Generic Dynamic Programming Matlab Function", In Proceedings of the 18th IEEE International Conference on Control Applications, pages 1625-1630, 2009.

[13] C. Godsil, G.F. Royle, "Algebraic Graph Theory", Springer, 2001.

[14] H. Rakha, R. K. Kamalanathsharma, "Eco-driving at signalized intersections using V2I communication", 14th International IEEE Conference on Intelligent Transportation Systems, 2011.

[15] M. Miyatake, M. Kuriyama, Y. Takeda, "Theoretical Study on Eco-Driving Technique for an Electric Vehicle Considering Traffic Signals", IEEE International Conference on Power Electronics and Drive Systems, 2011.