Technical report 14-015

Hybrid model predictive control for freeway traffic using discrete speed limit signals*

J.R.D. Frejo, A. Núñez, B. De Schutter, and E.F. Camacho

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

*This report can also be downloaded via https://pub.deschutter.info/abs/14_015.html

Hybrid model predictive control for freeway traffic using discrete speed limit signals

José Ramón D. Frejo[1], Alfredo Núñez,[2], Bart De Schutter,[3] and Eduardo F. Camacho[1]

Abstract

In this paper, two hybrid Model Predictive Control (MPC) approaches for freeway traffic control are proposed considering variable speed limits (VSL) as discrete variables as in current real world implementations. These discrete characteristics of the speed limits values and some necessary constraints for the actual operation of VSL are usually underestimated in the literature, so we propose a way to include them using a macroscopic traffic model within an MPC framework. For obtaining discrete signals, the MPC controller has to solve a highly non-linear optimization problem, including mixed-integer variables. Since solving such a problem is complex and difficult to execute in real-time, we propose some methods to obtain reasonable control actions in a limited computation time. The first two methods (θ-exhaustive and θ-genetic discretization) consist of first relaxing the discrete constraints for the VSL inputs; and then, based on this continuous solution and using a genetic or an exhaustive algorithm, to find discrete solutions within a distance θ of the continuous solution that provide a good performance. The second class of methods split the problem in a continuous optimization for the ramp metering signals and in a discrete optimization for speed limits. The speed limits optimization, which is much more time-consuming than the ramp metering one, is solved by a genetic or an exhaustive algorithm in communication with a non-linear solver for the ramp metering. The proposed methods are tested by simulation, showing not only a good performance, but also keeping the computation time reduced.

*The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement 257462 HYCON2 Network of excellence.

[1]J.R.D. Frejo and E.F. Camacho are with the Dept. de Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, University of Seville, Spain. {jdominguez3,eduardo}@us.es

[2]A. Núñez is with the Section of Road and Railway Engineering, Delft University of Technology, The Netherlands. a.a.nunezvicenciotudelft.nl

[3]B. De Schutter is with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands. b.deschuttertudelft.nl

I. INTRODUCTION

Traffic congestion on freeways is a critical problem due to its negative impact on the environment and many other important consequences like higher delays, waste of fuel, a higher accident risk probability, etc. In the last decades, a lot of research has been focused on making a better use of the available traffic infrastructure, since solutions like the construction of new freeways are not always viable to implement in the short-term due to technical, political, legal, or economic reasons. It has been reported in the literature under different conditions that dynamic traffic control is a good solution to decrease congestion [1]–[3]. In general, dynamic traffic control uses measurements of the traffic conditions over time and computes dynamic control signals to influence the behavior of the drivers and to generate a response in such a way that the performance of the network is improved, by reducing for example the delays, emissions, etc.

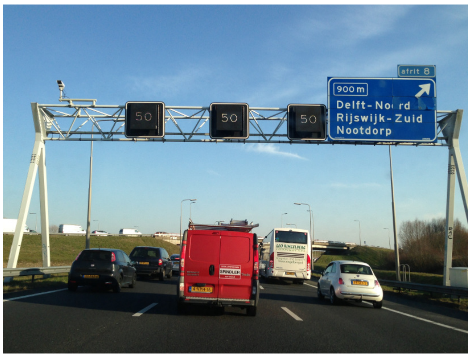

Fig. 1. VSL implementation on A13. Delft, The Netherlands.

Variable speed limits, ramp metering, and route guidance are some of the most often used examples of freeway traffic signals that can be used to dynamically control traffic. These measures have been already successfully implemented in USA, Germany, Spain, the Netherlands, and other countries [3]–[5]. When selecting the control signals, among the available options described in the literature, the methods based on the use of advanced control techniques like Model Predictive Control (MPC) [6] have by simulation proved to substantially improve the performance of the controlled traffic system [1], [2], [7], [8]. In the current paper, we focus on control using variable speed limits (VSL) together with ramp metering. In most of the works about VSL computed with MPC, the VSL signals are assumed to have continuous values, meaning that the real VSL panels implemented in the network should display to the driver those values [1], [2], [7], [9], [10]. However, in the real implementations of VSL panels, the displayed signals are just allowed to take a limited set of discrete values. For example, in the Dutch freeway A12 the signals of the panels are just allowed to take values in the set $\{60, 80, 100\}$km/h [3]. Moreover, for safety reasons, some extra constraints should be considered like a limited variation over time (for each panel) and a limited variation over space (consecutive panels) so to avoid drastic changes in speed [11], [12].

The following works have proposed ways to deal explicitly with discrete VSL. In [13], it is proposed a discrete VSL controller based on shock wave theory. In [14], a traffic model with variable length segments is used to compute a simple best-effort controller that reduces congestion considering VSL signals that can only be decreased or increased by steps of 10 km/h. Both controllers, [13] and [14], use simple control laws that are not explicitly designed to optimize a performance index of the network. In [11] and [12] the VSL are discretized (by rounding, ceiling, or flooring) after computing them in a continuous way. These papers conclude that the performance of the discretized speed limits was comparable with the continuous case. However, those results depend on the network configuration and the demand conditions, as in our case study we found some important loss of performance due to the discretization. So, in this paper, we consider explicitly the effects of using discrete signals for the VSL panels in an MPC framework. Subsequently, we propose new efficient algorithms for the computation of discrete VSL signals together with continuous ramp metering rates.

Section II briefly introduces the general concepts of traffic model METANET, of Model Predictive Control, and the safety and operational constraints that appear in a real VSL implementation. Section III introduces the common characteristics of the proposed methods to obtain discrete signals using the continuous solution for the VSL panels. In Section IV, an exhaustive and a genetic procedure are used to obtain feasible discrete solutions within a θ distance to the continuous solution provided by the MPC controller. Section V proposes methods to obtain the MPC solution by solving iteratively a continuous optimization for the ramp metering and a discrete optimization for the VSL without using the continuous MPC solution. Again, an exhaustive algorithm and a genetic algorithm are proposed. The scenario and the numerical results are presented and discussed in Section VI. Finally, Section VII shows a summary of the results and the conclusions.

II. FREEWAY TRAFFIC CONTROL USING MODEL PREDICTIVE CONTROL

A. Traffic Model METANET

In this paper, we have selected the traffic model METANET [15]. However, it is important to note that the methods we propose are independent of the traffic model used, so they can be equivalently applied using other macroscopic traffic models, if those are capable of including the effect of VSL in their formulation (like some versions of the Cell Transmission Model CTM [16]).

The METANET model is a macroscopic second-order traffic model that provides a good trade-off between simulation speed and accuracy for online control purposes. The METANET model is deterministic and can be adapted to freeway networks of arbitrary topology and characteristics, including freeway stretches, bifurcations, on-ramps, and off-ramps, and it takes into account the effects of control actions such as ramp metering, route guidance, and VSL. This model discretizes the freeway in consecutive links with segments of length L_i and uses density $\rho_i(k_\mathrm{m})$ and speed $v_i(k_\mathrm{m})$ as state variables where k_m is the model time step. For simplicity, in this paper the authors do not differentiate between links and segments in contrast with the original METANET model. The model is formulated as follows:

- ***Density equation:***

$$\rho_i(k_\mathrm{m}+1) = \rho_i(k_\mathrm{m}) + \frac{T_\mathrm{m}}{\lambda_i L_i}(q_{i-1}(k_\mathrm{m}) - q_i(k_\mathrm{m}) + q_{\mathrm{r},i}(k_\mathrm{m}) - \beta_i(k_\mathrm{m})q_{i-1}(k_\mathrm{m})) \tag{1}$$

where λ_i is the number of lanes, $\beta_i(k_\mathrm{m})$ is the split ratio for off-ramp in between segment i and segment $i+1$ ($\beta_i(k_\mathrm{m}) = 0$ if there is not an off-ramp), T_m is the model sample time, $q_i(k_\mathrm{m})$ is the flow leaving segment i, and $q_{\mathrm{r},i}(k_\mathrm{m})$ is flow entering by an on-ramp at the start of segment i ($q_{\mathrm{r},i}(k_\mathrm{m}) = 0$ for a segment without an on-ramp).

- ***Speed equation:***

$$\begin{aligned} v_i(k_\mathrm{m}+1) = v_i(k_\mathrm{m}) + \frac{T_\mathrm{m}}{\tau_i}\left(V(\rho_i(k_\mathrm{m})) - v_i(k_\mathrm{m})\right) + \frac{T_\mathrm{m}}{L_i} v_i(k_\mathrm{m})(v_{i-1}(k_\mathrm{m}) - v_i(k_\mathrm{m})) \\ - \frac{\mu_i T_\mathrm{m}}{\tau_i L_i}\frac{\rho_i(k_\mathrm{m}+1) - \rho_i(k_\mathrm{m})}{\rho_i(k_\mathrm{m}) + K_i} - \frac{\delta_i T_\mathrm{m} q_{\mathrm{r},i}(k_\mathrm{m}) v_i(k_\mathrm{m})}{L_i \lambda_i(\rho_i(k_\mathrm{m}) + K_i)} \end{aligned} \tag{2}$$

where K_i, τ_i, δ_i, and μ_i are model parameters and $V(\rho_i(k_\mathrm{m}))$ is the speed desired for the drivers. As proposed in [1], the model can take different values for μ_i, depending on whether the downstream density is higher or lower than the density in the corresponding segment. However, the algorithms we propose are independent of this choice and could be applied in both cases.

- ***Flow equation:***

$$q_i(k_\mathrm{m}) = \lambda_i \rho_i(k_\mathrm{m}) v_i(k_\mathrm{m})$$

- ***Desired speed equation:***

$$V(\rho_i(k_\mathrm{m})) = \min\left(v_{\mathrm{f},i} \exp\left(-\frac{1}{a_i}\left(\frac{\rho_i(k_\mathrm{m})}{\rho_{\mathrm{c},i}}\right)^{a_i}\right), (1+\alpha_i)V_{\mathrm{c},i}(k_\mathrm{m})\right) \tag{3}$$

where a_i, α_i are model parameters, $\rho_{\mathrm{c},i}$ is the critical density, $v_{\mathrm{f},i}$ is the free flow speed and $V_{\mathrm{c},i}(k_\mathrm{m})$ is the variable speed limit applied to segment i.

In this paper, the VSL are included by a minimum term in the desired speed equation (3) as proposed in [1]. However, in [15], VSL are included in the model through the three parameters of the fundamental diagram $\rho_{\mathrm{c},i}$, $v_{\mathrm{f},i}$ and a_i. In [17] the effect of VSL on aggregate traffic flow behavior is studied on the basis of traffic data comparing the equation (3) used in this paper with other options. The methods proposed in this paper are independent from the option selected.

If the desired flow $Q(\rho_i(k)) = V(\rho_i(k)) \cdot \rho_i(k)$ is represented graphically without considering VSL, the fundamental diagram (which can be seen in Fig.2.) of traffic systems is obtained. The fundamental diagram gives us the static characteristic of the system.

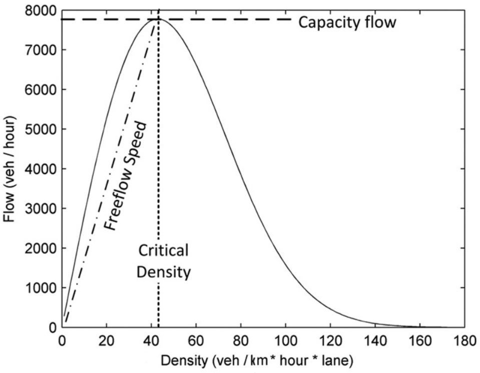

Fig. 2. Fundamental diagram of traffic

- ***Ramp flow equation:***

$$q_{\mathrm{r},i}(k_\mathrm{m}) = \min\left(r_i(k_\mathrm{m})C_{\mathrm{r},i}, D_i(k_\mathrm{m}) + \frac{w_i(k_\mathrm{m})}{T_\mathrm{m}}, C_{\mathrm{r},i}\frac{\rho_{\mathrm{m},i} - \rho_i(k_\mathrm{m})}{\rho_{\mathrm{m},i} - \rho_{\mathrm{c},i}}\right) \tag{4}$$

where $C_{\mathrm{r},i}$ is the ramp capacity, $D_i(k_\mathrm{m})$ is the ramp demand, $w_i(k_\mathrm{m})$ is the ramp queue length, $\rho_{\mathrm{m},i}$ is the maximum density, and $r_i(k_\mathrm{m})$ is the ramp metering rate.

- ***Queue length equation:***

$$w_i(k_\mathrm{m}+1) = w_i(k_\mathrm{m}) + T_\mathrm{m} \cdot (D_i(k_\mathrm{m}) - q_{\mathrm{r},i}(k_\mathrm{m})) \tag{5}$$

For the sake of simplicity, merge and join nodes, and other extensions are not considered in this paper (See [15], [18] for further details).

B. Model Predictive Control

The main concepts behind a model-based predictive control (MPC) [6] strategy are:
1) The use of a prediction model to obtain the trajectories of relevant variables of the system.
2) The optimization of a objective function to determine the best sequence of control actions for the system.
3) The application of the rolling horizon procedure: from the best sequence of control actions only the first component is applied to the system and in the next control step the initial conditions are updated and the procedure is repeated again.

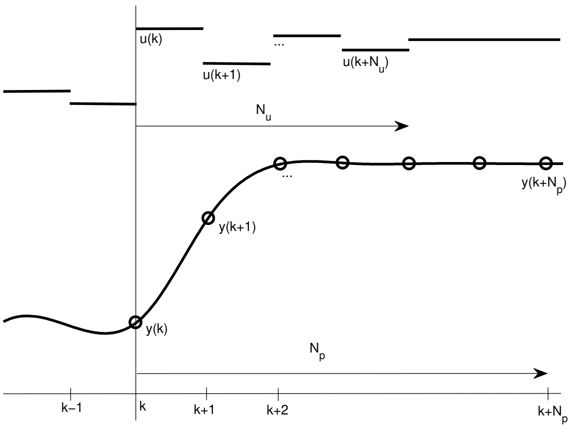

Fig. 3. Receding horizon strategy. The set of future control signals are computed considering the predicted outputs during the prediction horizon, but just the first control $u(k)$ is applied

Since MPC is based on an optimization procedure, in the case of systems with discrete control actions we can include that characteristic as a constraint, and obtain suitable control laws by using a proper mixed-integer optimization solver.

To formalize all these concepts, consider the discrete-time non-linear system whose dynamic evolution is described by the following state-space model:

$$x(k_{\mathrm{m}}+1) = f(x(k_{\mathrm{m}}), u(k_{\mathrm{m}}), d(k_{\mathrm{m}})) \tag{6}$$

with $x(k_{\mathrm{m}})$ the state, $u(k_{\mathrm{m}}) = [(u^{\mathrm{d}}(k_{\mathrm{m}}))^T, (u^{\mathrm{c}}(k_{\mathrm{m}}))^T]^T$ the discrete and continuous input vector, and $d(k_{\mathrm{m}})$ the non-controllable input vector, usually demand profiles and other exogenous variables. In some circumstances, the controller sample time may be different from the model sample time. In traffic we will use a controller sample time T longer than the simulation sample time T_{m} (i.e. $t = k_{\mathrm{m}} \cdot T_{\mathrm{m}} = k \cdot T$). The control inputs will be considered constant during one controller sample resulting in the following state-space model:

$$x(k+1) = f(x(k), u(k), d(k)) \tag{7}$$

In an MPC controller, the core is the optimization of a cost function $J(x_{\mathrm{t}}(k), u_{\mathrm{t}}(k), d_{\mathrm{t}}(k))$, which is used to measure the performance of the system where the vectors $x_{\mathrm{t}}(k) = [x(k+1), ..., x(k+N_{\mathrm{p}})]^T$ and $d_{\mathrm{t}}(k) = [d(k), ..., d(k+N_{\mathrm{p}}-1)]^T$ are the state and non-controllable input predictions along the prediction horizon N_{p}; and $u_{\mathrm{t}}(k) = [u(k), ..., u(k+N_{\mathrm{u}}-1)]^T$ is the control sequence along the control horizon N_{u}. The control inputs are kept constant after the control horizon N_{u} (i.e. $u(k+N_{\mathrm{u}}-1) = u(k+N_{\mathrm{u}}) = ... = u(k+N_{\mathrm{p}}-1)$).

As the formulation is based on the solution of an optimization problem, it is possible to explicitly include constraints. Assume that the feasible values of the states and the inputs are given by the following generic constraints $x(k) \in \mathbb{X}$, $u^{\mathrm{c}}(k) \in \mathbb{U}$, and $u^{\mathrm{d}}(k) \in \mathbb{S}$ for all k, representing explicitly physical or operational constraints of the system. Therefore, the MPC problem can be formulated as the following mixed-integer non-linear optimization problem:

$$\min_{u_{\mathrm{t}}(k)} J(u_{\mathrm{t}}(k), x_{\mathrm{t}}(k), d_{\mathrm{t}}(k)) \tag{8}$$

subject to:

$$x(k+\ell+1) = f(x(k+\ell), u(k+\ell), d(k+\ell)), \quad x(k) = x_k$$

$$x(k+\ell+1) \in \mathbb{X}, \qquad u^{\mathrm{c}}(k+\ell) \in \mathbb{U}, \qquad u^{\mathrm{d}}(k+\ell) \in \mathbb{S},$$

$$h(u_{\mathrm{t}}(k), x_{\mathrm{t}}(k), d_{\mathrm{t}}(k)) \in \mathbb{D},$$

$$\text{for } \ell = 0, 1, ..., N_{\mathrm{p}}-1,$$

with $\mathbb{S}$ the set of possible control values $\mathbb{S} = \{u_1, u_2, ..., u_M\}$ for the corresponding feasible discrete input components (in our case, the VSL). The term $h(u_t(k), x_t(k), d_t(k)) \in \mathbb{D}$ corresponds to the rest of the constraints and x_k is last measured state at time step k. Using the rolling horizon procedure, only the first control action $u(k)$ of the optimal sequence is applied to the system, and in the next time step the initial conditions are updated and the procedure is repeated.

To properly consider processes with discrete and continuous variables (hybrid systems) in an MPC formulation, hybrid predictive control techniques have been developed [19]–[21]. For this formulation, the main difficulty is the computation time needed to solve the optimization problem because we are dealing with mixed integer nonlinear programming (MINLP). In cases when the problem can be recast into a MILP, well-known optimization methods are efficient and many software/toolboxes are available to solve them [22]. However, in the case of traffic systems, the problem is intrinsically non-linear and any simplification of the model may lead to non-acceptable predictions, which are incapable of incorporating the real behavior of traffic. As a way to deal with complexity, and to include the discrete characteristic explicitly in the solution, we propose a set of methods based on genetic algorithms.

C. Particularities of the MPC controllers used in this paper:

This subsection explains the main particularities of the MPC controllers used in this paper:
–The MPC controller uses an objective function (9) containing one term for the Total Time Spent (TTS) and two terms that penalize abrupt variations in the ramp metering and VSL:

$$J(k) = \sum_{\ell=1}^{N_\mathrm{p}} T[\ \sum_{i\in\mathbb{O}} w_i(k+\ell)\ +\ \sum_{li\in\mathbb{I}} (\rho_i(k+\ell)L_i\lambda_i)\ ] \tag{9}$$

$$+\ \sum_{\ell=0}^{N_\mathrm{u}-1} \epsilon_\mathrm{vc} \parallel V_{\mathrm{c},j}(k+\ell) - V_{\mathrm{c},j}(k+\ell-1) \parallel^2 \tag{10}$$

$$+\ \sum_{\ell=0}^{N_\mathrm{u}-1} \epsilon_\mathrm{r} \parallel r_{\mathrm{c},j}(k+\ell) - r_{\mathrm{c},j}(k+\ell-1) \parallel^2$$

where ϵ_vc, ϵ_r are weighting parameters, $\mathbb{O}$ is the set of all the segments with an on-ramp, and $\mathbb{I}$ the set of all the segments.
–The controlled system is subject to constraints on the maximum and minimum values of densities, speeds, queues, ramp metering rates, and VSL. The constraints on speed, density, and queue are made soft by including them as penalization terms in the overall cost function:

$$\bar{J}(k) = J(k) + \sum_{\ell=1}^{N_\mathrm{p}} \sum_{i=1}^{N_\mathrm{const}} \Omega_i(k+\ell) \tag{11}$$

where $\Omega_i(k+\ell)$ is a penalization term that is different to zero if the corresponding soft constraint is violated and N_const is the total number of soft constraints.
–The optimization is computed using SQP optimization techniques by the Matlab function fmincon with a control and prediction horizon of $N_\mathrm{u} = 4$ and $N_\mathrm{p} = 6$, respectively. The function uses the interior point method with a maximum number of 20000 cost function evaluations.
–In order to try to avoid that the algorithm ends up in a local minimum, the algorithm runs

an evaluation procedure before the optimization. During the procedure, the TTS is evaluated for a set of control values (in this case, 252 points). The unfeasible candidates with a small cost function associated are used for the definition of a new feasible candidate increasing the set of control values. The best control profile obtained is taken as initial values for the optimizations. The right choice of this set of initial values is a key step in the design of a continuous MPC for traffic due to the high non-convexities of the optimization problem. To use the shifted version of that part of the control signal that was not applied to the process or the minimum feasible metering rates and VSL (as proposed in [1]) is not enough and may cause high loss of performance for the majority of the simulations done in this paper. On the other hand, to use a multi-start optimization increases the complexity making, in many cases, impossible to compute the control inputs in real time if the optimizations cannot be solved in parallel.

D. VSL Implementation Issues

In the majority of the references about VSL computed by MPC, the variable speed limits are considered to be implemented in the network without any operational constraints. Hereafter, we will assume the MPC controllers will have to consider the following safety and operational constraints for the VSL:

- ***Temporal constrained case:***

In real implementations, the VSL cannot change abruptly due to driver safety and comfort. Therefore, we just allow the VSL to change γ_i km/h as maximum in a controller sample step including the following constraint as in [11], [12]:

$$|V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{c},i}(k+\ell) - 1| \leq \gamma_i \tag{12}$$

for each segment i with a VSL and for $\ell = 0, 1, ...N_{\mathrm{u}} - 1$.

Hereafter, the MPC proposed with temporally constrained VSL will be denoted by T-MPC. Unlike the state constraints, the constraint (12) is treated as hard constraint.

- ***Spatial and temporal constrained case:***

In addition, in real implementations it is necessary to limit the difference between the VSL values of two adjacent segments. This can be done by including the following hard constraint as in [11], [12]:

$$|V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{c},i}(k+\ell) - 1| \leq \gamma_i \tag{13}$$

for each segment and i with a VSL andfor $\ell = 0, 1, ...N_{\mathrm{u}} - 1$.

$$|V_{\mathrm{c},j+1}(k+\ell) - V_{\mathrm{c},j}(k+\ell)| \leq \zeta_j$$

for j and $j+1$ corresponding to adjacent VSL and for $\ell = 0, 1, ...N_{\mathrm{u}} - 1$.

Hereafter, the MPC proposed with spatially and temporally constrained VSL will be denoted by ST-MPC.

- ***Discrete case:***

Even with the previously proposed ST-MPC, the solution obtained cannot be implemented in practice since this solution is continuous and the freeway signs are only allowed to show a limited set $\mathbb{S}$ of discrete speed limits. An easy but suboptimal method is to approximate the continuous optimal control inputs by allowed VSL values (i.e. to apply a discretization process). The most used way to do this discretization is by rounding the available continuous solution to the closest discrete value of the VSL [11], [12]:

$$V_{\mathrm{c},i}(k) = \arg\min_{s \in \mathbb{S}}(|V_{\mathrm{cST},i}(k) - s|) \tag{14}$$

where $V_{\mathrm{cST},i}(k)$ is the VSL of segment i computed by ST-MPC controller explained in the previously.

Other easy ways to do the discretization are by ceiling (15), and by flooring (16):

$$V_{\mathrm{c},i}(k) = \arg\min_{s \in \mathbb{S}}(|V_{\mathrm{cST},i}(k) - s| \text{ subject to: } s \geq V_{\mathrm{cST},i}(k)) \tag{15}$$

$$V_{\mathrm{c},i}(k) = \arg\min_{s \in \mathbb{S}}(|V_{\mathrm{cST},i}(k) - s| \text{ subject to: } s \leq V_{\mathrm{cST},i}(k)) \tag{16}$$

In [1], it is reported that the discretization done by ceiling yields a slightly better performance than the rounding and the flooring case. In Section IV alternative methods for the discretization step will be proposed in order to decrease the loss of performance due to the quantization.

III. HYBRID FREEWAY TRAFFIC CONTROL

A. Problem Definition

The implementation issues explained in the previous section substantially decrease the performance of the controlled system as will be shown in the simulation results in Section

V and VI. In order to reduce this loss of performance, the discrete characteristic of the VSL values should be directly considered during the optimization by solving or approximating the following hybrid optimization problem:

$$\begin{aligned} \min_{r_{\mathrm{t}}(k), V_{\mathrm{c,t}}(k)} \quad & \bar{J}(k) \\ \text{s.t:} \quad & 0 \leq r(k+\ell) \leq 1 \\ & \mathrm{VSL}_{\min} \leq V_{\mathrm{c}}(k+\ell) \leq \mathrm{VSL}_{\max} \\ & \text{for } \ell = 0, 1, ..., N_{\mathrm{u}} - 1 \\ & \text{with } r_{\mathrm{t}}(k) \in \mathbb{R}^{\mathbb{N}_{\mathrm{u}} \cdot \mathbb{N}_{\mathrm{r}}} \text{ and } V_{\mathrm{c},t}(k) \in \mathbb{S}^{\mathbb{N}_{\mathrm{u}} \cdot \mathbb{N}_{\mathrm{VSL}}} \end{aligned} \tag{17}$$

where $r_{\mathrm{t}}(k) = [r(k)^T, ..., r(k+N_{\mathrm{u}}-1)^T]^T$, and $r(k) = [r_{j_1}(k), ..., r_{j_{N_{\mathrm{r}}}(k)}]^T$ with $\mathbb{I}_{\mathrm{r}} = \{j_1, j_2, ..., j_{N_{\mathrm{r}}}\}$ the set of segments with a metered on-ramps and N_{r} the number of metered on-ramps. Equivalently, $V_{\mathrm{c},t}(k)$ is the discrete control input profile: $V_{\mathrm{c,t}}(k) = [V_{\mathrm{c}}(k)^T, ..., V_{\mathrm{c}}(k+N_{\mathrm{u}}-1)^T]^T$, and $V_{\mathrm{c}}(k) = [V_{\mathrm{c},m_1}(k), ..., V_{\mathrm{c},m_{N_{\mathrm{VSL}}}(k)}]^T$ with $\mathbb{I}_{\mathrm{VSL}} = \{m_1, m_2, ..., m_{N_{\mathrm{r}}}\}$ the set of segments with a VSL and N_{VSL} the number of VSL.

B. Problem Relaxation

The proposed optimization problem (17) is a complex to solve mixed-integer optimization problem with discrete and continuous decision variables and so it is difficult to execute in real time. Most of the MPC approaches for hybrid systems do not have a standard strategy to relax a non-convex problem in order to obtain a good solution in a reasonable amount of computation time. Limiting the number of feasible nodes in the mixed integer optimization problem is a strategy that has been used before in the context of MPC for sewer networks [23], as a way to reduce the combinatorial explosion related to the search tree of binary solutions at the expense of a suboptimal solution. In this paper, a number of feasible nodes limitation will be obtained by using the spatial and temporal implementation constraints explained in subsection II.C:

$$\begin{aligned} & |V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{c},i}(k+\ell-1)| \leq \gamma_i \\ & |V_{\mathrm{c},j+1}(k+\ell) - V_{\mathrm{c},j}(k+\ell-1)| \leq \zeta_j \\ & \text{for } \ell = 0, 1, ...N_{\mathrm{u}} - 1, \text{ for } i \in \mathbb{I}_{\mathrm{VSL}}, \text{ and for } j \text{ and } j+1 \in \mathbb{I}_{\mathrm{VSL}} \end{aligned} \tag{18}$$

The feasible VSL set can be represented by a search tree. Fig. 4 shows an example of a search tree with 2 VSL, $N_{\mathrm{u}} = 2$, $V_{\mathrm{c}}(k) = [40, 50]^T$, $\zeta_i = 10$, $\gamma_i = 10$, and $\mathbb{S} =$

$\{20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\}$ where the unfeasible VSL profiles have already been removed:

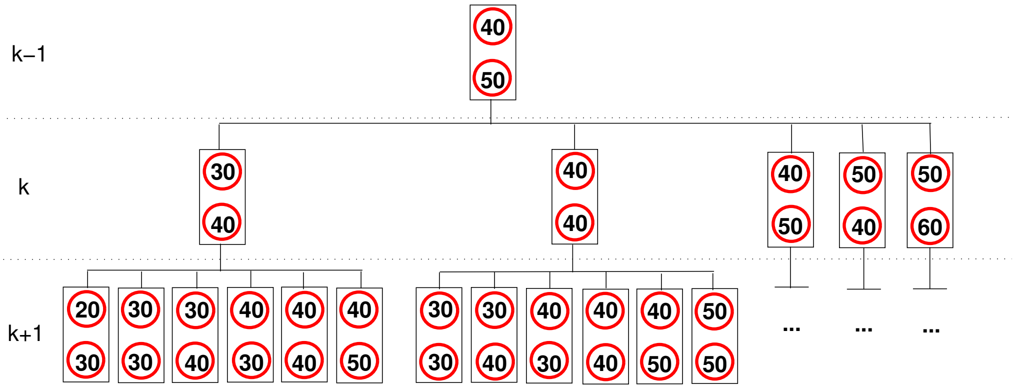

Fig. 4. Search tree example with $N_u = 2$ reduced by the spatial and temporal constraints.

In this case, the temporal constraints reduce the number possible combinations of discrete VSL from 1331 to only 81. The spatial constraint reduces again this number from 81 to 38 profiles of discrete VSL. For larger horizons, the number of feasible VSL profiles should increase exponentially. For example, in the simulation done in this paper (with $N_{\text{u}} = 4$ and two VSL) the search tree is initially composed of $2.14 \cdot 10^8$ possible VSL profiles. However, this number is reduced from $2.14 \cdot 10^8$ to 6561 VSL profiles by the temporal constraint and from 6561 to an average number of 1998 VSL profiles by the spatial constraint.

In the following two sections, some methods will be proposed in order to solve optimization (17) with constraints (18) using the search tree of Fig. 4.

IV. θ-GENETIC AND θ-EXHAUSTIVE OPTIMIZATIONS

The first proposed methods makes use of the continuous solution of the ST-MPC problem (Section II.C) in order to relax problem (17) with constraints (18). The idea is to run a discrete optimization after the continuous optimization assuming that the optimal discrete solution is close to the continuous one. This is not generally true and so, in some cases, could cause loss of optimality.

Firstly, the VSL search tree explained in the previous section is reduced again by including a new constraint in the optimization problem making use of the ST-MPC solution for the

VSL ($V_{\mathrm{cST},i}(k)$):

$$|V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{cST},i}(k+\ell)| \leq \theta \tag{19}$$
$$\text{for } \ell = 0, 1, ...N_{\mathrm{u}} - 1 \text{ and for } i \in \mathbb{I}_{\mathrm{VSL}}$$

where θ is a tuning parameter.

Therefore, the feasible VSL set can be represented by a smaller search tree than in Section III (Fig. 4). In Fig. 5 an example of this search tree with $V_{\mathrm{cST}}(k+1) = [43, 53]^T$, $V_{\mathrm{cST}}(k+1) = [52, 61]^T$ and $\theta = 10$ is shown. In this case, the constraints reduce the number possible combinations of discrete VSL from 1331 to only 6 (versus 38 without using constraint (19)).

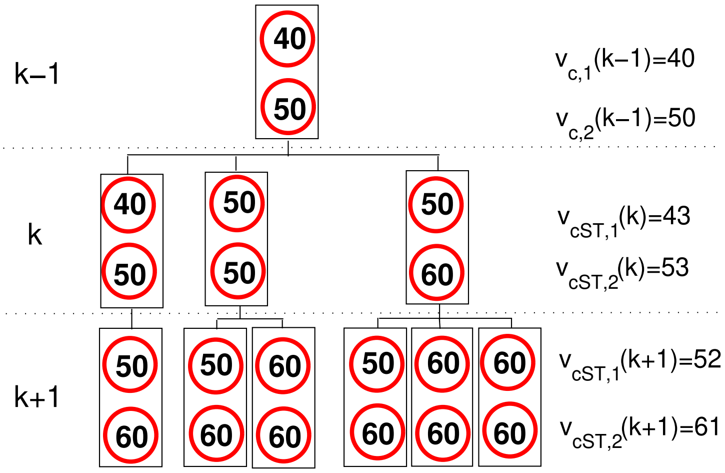

Fig. 5. Search tree of Fig. 4 using the ST-MPC solution

For the simulation done in this paper (with $N_{\mathrm{u}} = 4$ and $\theta = 10$) the search tree is reduced from $2.14 \cdot 10^8$ points to an average number of 114 points (versus 1998 without constraint (19)). If θ is chosen to be 14, the average number of points in the reduced tree is 971 because the larger the θ, the more feasible solutions there are to explore.

Moreover, making use of the ST-MPC solution for the ramp metering rates $r_{\mathrm{cST},j}(k)$, the optimization can be reduced to an optimization with only discrete variables. Therefore, in order to obtain the discrete speed limits using these assumptions (given ramp metering rate and constraint (19)), the following optimization problem with only discrete variables has to be solved:

$$\min_{V_{\mathrm{c,t}}(k)} \quad \bar{J}(k) \tag{20}$$

$$\text{s.t:} \quad |V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{c},i}(k+\ell-1)| \leq \gamma_i, \quad |V_{\mathrm{c},j+1}(k+\ell) - V_{\mathrm{c},j}(k+\ell)| \leq \zeta_j,$$

$$|V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{cST},i}(k+\ell)| \leq \theta \quad \text{and} \quad \mathrm{VSL}_{\mathrm{min}} \leq V_{\mathrm{c},i}(k+\ell) \leq \mathrm{VSL}_{\mathrm{max}}$$

$$\text{for } \ell = 0, 1, ...N_{\mathrm{u}} - 1, \text{ for } i \in \mathbb{I}_{\mathrm{VSL}}, \text{ and for } j \text{ and } j+1 \in \mathbb{I}_{\mathrm{VSL}} \text{ with } V_{\mathrm{c},i}(k+\ell) \in \mathbb{S}$$

A. θ-Exhaustive optimization

As we are dealing with non-convex integer optimization, the only way to obtain the global optimum of (20) is to evaluate the cost function for all the feasible points in the reduced search tree. The main problem is the computation time needed for the evaluation of such a large number of possible combinations of discrete VSL. Therefore, this solution is just applicable for relatively small networks and horizons or in cases where an offline solution is useful. In order to be able to solve the problem for large networks within the limited computation time available, a genetic algorithm is proposed in the following section.

B. θ-Genetic optimization

To optimize the discrete control action sequences $V_{\mathrm{c},i}(k)$ (i.e. to solve problem (20)) within the limited computation time, the use of Genetic Algorithms (GA) [24] is proposed. In cases where the optimization problem is highly nonlinear, it has been reported that GA can efficiently cope with it, particularly for mixed-integer nonlinear problems [24]. In the context of MPC, in cases when the model or the objective function are nonlinear, the use of GA is fully justified as the computation times can be easily controlled and good solutions can be obtained within a fixed sampling time [25]–[27]. One of the main characteristics of this method is that it is gradient-free and that it is possible to limit the number of objective function evaluations. So, in order to find a good discrete VSL within the limited time available, a GA will be used to solve optimization problem (20).

A candidate control sequence solution in the genetic algorithm is called an individual, and each individual has a fixed number of genes (emulating a chromosome). Each gene, in the context of systems with discrete inputs, will represent a possible input during the control horizon. For MPC, the individual can be seen then as a candidate control action sequence:

$$\mathrm{Individual}_1 = [V_{\mathrm{c},m_1}(k), ..., V_{\mathrm{c},,m_1}(k+N_{\mathrm{u}}-1), ..., \tag{21}$$
$$V_{\mathrm{c},m_{N_{\mathrm{VSL}}}}(k), ..., V_{\mathrm{c},m_{N_{\mathrm{VSL}}}}(k+N_{\mathrm{u}}-1)]$$

where $\{m_1, m_2, ..., m_{N_\mathrm{r}}\}$ is the set of segments with a VSL as explained previously.

In GA the idea is to find the fittest individuals (solutions with best objective function values) within a generation, to apply genetic operators for the recombination of those individuals, and to generate a good offspring [24]. In this paper, for the selection, a roulette method is applied, giving the best individuals more chances to be selected for recombination. For the recombination, two fundamental operators are used: crossover and mutation. For the crossover, portions of the chromosomes of two individuals are exchanged with a given probability p_c; and the mutation operator modifies each gene randomly with a given probability p_m. This is just an specific implementation and alternative genetic methods could be used.

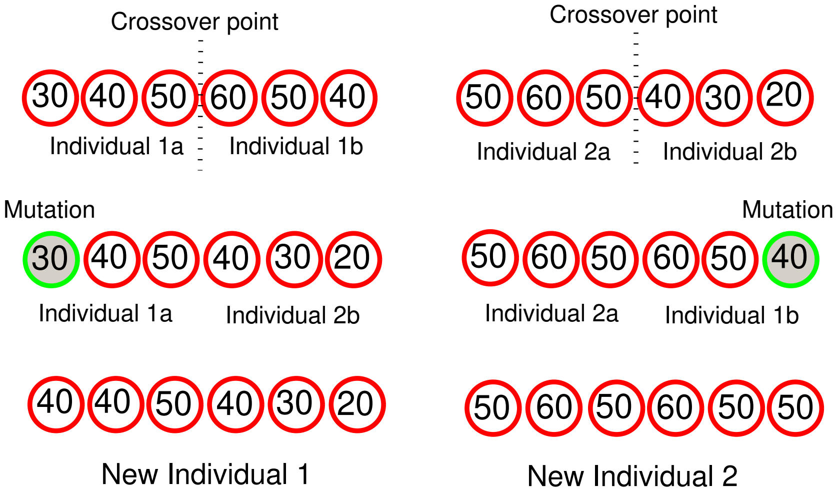

Fig. 6. The basic operators in a GA-based control strategy for VSL panels.

Fig. 6 shows an example of the recombination steps for two individuals assuming that we have two VSLs and the control horizon is 3. First two individuals are selected for recombination with a higher probability if their objective function is lower. Then, a random crossover point is selected, and two new individuals are generated. Then the mutation operator selects randomly a control to modify and its values are changed. Finally two new individuals for a new generation are obtained.

We have to point out that due to the limited time reaching the global optimum is not guaranteed. Since for traffic control the optimization is a complex mixed-integer and nonlinear problem, using the GA optimization is justified. Many different approaches and adaptations of genetic algorithms have been proposed in the literature in order to deal with many issues like constraint handling, diversity of the solutions, combination with classical optimization methods to assure a local convergence, etc [28]–[31]. For constrained optimization, one of the most used is the GENOCOP algorithm proposed by Michakewicz [32]. The algorithm we used for the traffic application is the simplest one [24] and in the case a solution does not satisfy a constraint, we penalize with a high objective function value. Part of the further research is to implement different adaptations of GA, and other algorithms. In the simulation results we will compare GA with exhaustive enumeration, so we can see how far the solutions of GA are from the optimal solution.

V. ALTERNATING OPTIMIZATION

In the previous section, the continuous VSL given by a MPC controller have been used for the computation of the discrete ones. However, these discrete VSL could be computed directly (i.e. without using any continuous solution) saving the computation time needed for the continuous optimization (i.e. to solve optimization (17) without using the ST-MPC solution). To solve the mixed-integer optimization problem with discrete and continuous decision variables, we decided to compute the ramp metering and the VSL iteratively in order to decompose the problem in one continuous optimization problem (22) and one discrete optimization problem (23).

1) Ramp metering optimization:

$$\min_{r_{\mathrm{t}}(k)} \quad \bar{J}(k) \tag{22}$$

$$\text{s.t:} \quad 0 \leq r_i(k+\ell) \leq 1 \text{ for } i \in \mathbb{I}_{\mathrm{r}} \text{ , for } \ell = 0, 1, ...N_{\mathrm{u}} - 1 \text{ and with } r_t(k) \in \mathbb{R}^{\mathrm{N_u \cdot N_r}}$$

where the VSL profile used $V^*_{\mathrm{c},t}(k)$ for the ramp metering optimization is the profile proposed in the previous iteration or, for the first iteration, the profile proposed in the previous sample time shifted by one sample. In this paper, this continuous ramp metering optimization problem will be solved using an SQP algorithm. The computation time needed is much lower than the computation time needed for the general continuous optimization problem because it does not include the VSL signals as decision variables.

2) VSL optimization:

$$\begin{aligned}
&\min_{V_{\mathrm{c,t}}(k)} \quad \bar{J}(k) && (23)\\
&\text{s.t:} \quad |V_{\mathrm{c},i}(k+\ell) - V_{\mathrm{c},i}(k+\ell-1)| \leq \gamma_i \quad |V_{\mathrm{c},j+1}(k+\ell) - V_{\mathrm{c},j}(k+\ell)| \leq \zeta_j \\
&\qquad \mathrm{VSL}_{\min} \leq V_{\mathrm{c},i}(k+\ell) \leq \mathrm{VSL}_{\max} \\
&\qquad \text{for } \ell = 0, 1, \ldots N_{\mathrm{u}} - 1, \text{ for } i \in \mathbb{I}_{\mathrm{VSL}}, \text{ and for } j \text{ and } j+1 \in \mathbb{I}_{\mathrm{VSL}} \text{ with } V_{\mathrm{c},i}(k+\ell) \in \mathbb{S}
\end{aligned}$$

where the ramp profile used is the one proposed in the previous iteration $r_t^*(k)$. This optimization (23) is equivalent to the optimization done for the previous section (20) but, in this case, we do not have any continuous MPC solution for the VSL that can be used for the reduction of the search space of the problem. The following subsections propose some ways to solve the resulting discrete optimization problem (23), as done for the problem (20) in Section IV.

A. Exhaustive optimization

In the same way that it was explained in Section IV, the easiest way to solve problem (23) is to evaluate the cost function for all the feasible points. Again, the problem is the computation time needed for the evaluation of such a large number of possible combinations of discrete VSL. Therefore, this solution is just applicable for relatively small networks and horizons. For larger networks and horizons, a genetic algorithm is proposed in the following subsection.

B. Genetic optimization

In the same way that in Section IV.B, a genetic algorithm is proposed to solve the discrete optimization. The formulation is the same as in Section IV but without using the constraint related with a solution within a distance θ from the continuous one. Moreover, it will be convenient to use a different GA parameters tuning due the greater number of feasible solutions and the larger computation times available.

VI. SIMULATION

A. Set-up and scenario

In order to simulate the analyzed controllers, the benchmark network in Fig. 7 has been used. The network has been taken from [1]. The freeway has $N = 6$ segments with a length of $L_i = 1000$ m and with $\lambda = 2$ lanes. There are 3 control signals: two VSL (for segments 3 and 4) and a ramp metering rate. All the METANET parameters (which can be seen in Table I) are considered to be the same for all the segments.

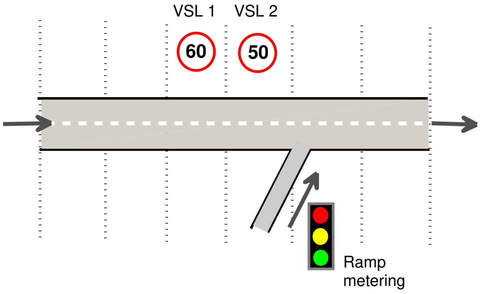

Fig. 7. Stretch used as example.

TABLE I

MODEL PARAMETERS

τ	18 s
K	40 veh/(km· lane)
ρ_{crit}	33.5 veh/(km· lane)
α	0.1
a	1.867
v_{free}	102 km/h
μ	60 km^2/h
δ	0.0122
$\rho_{\max}$	180 veh/(km· lane)
C_{o}	4000 veh/h
C_{ramp}	2000 veh/h

Thirteen variables (density and speed of each segment and queue of the ramp) are supposed to be measured at each controller sample step and used for the computation of the control signals. The simulation time chosen is two and half hour corresponding to 75 controller sample steps ($T = 120\text{s}$) and 900 simulation steps ($T_{\text{m}} = 10\text{s}$). See [1] [8] for further details about the chosen benchmark. The set of allowed VSL is supposed to be $\mathbb{S} = \{20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\}$ and the implementation constraints parameters are $\zeta_i = 10$, and $\gamma_i = 10$.

B. Continuous MPC controllers results

The results of the closed-loop simulations for the VSL computed by each continuous MPC controller and the rounding discretization of ST-MPC can be seen in Fig. 8. It can be seen that VSL_1 of unconstrained MPC suddenly changes from 76 km/h to 32 km/h at minute 12. However, this change is done at a rate of 10 km/h for T-MPC. Moreover, the absolute difference between VSL_1 and VSL_2 (i.e. $|\text{VSL}_1 - \text{VSL}_2|$) keeps between 40 km/h and 60 km/h for unconstrained MPC and T-MPC during a long period of time. When this difference is constrained in ST-MPC, the value of VSL_2 is strongly decreased in order to obtain a difference of 10 km/h with VSL_1 (which is also slightly increased).

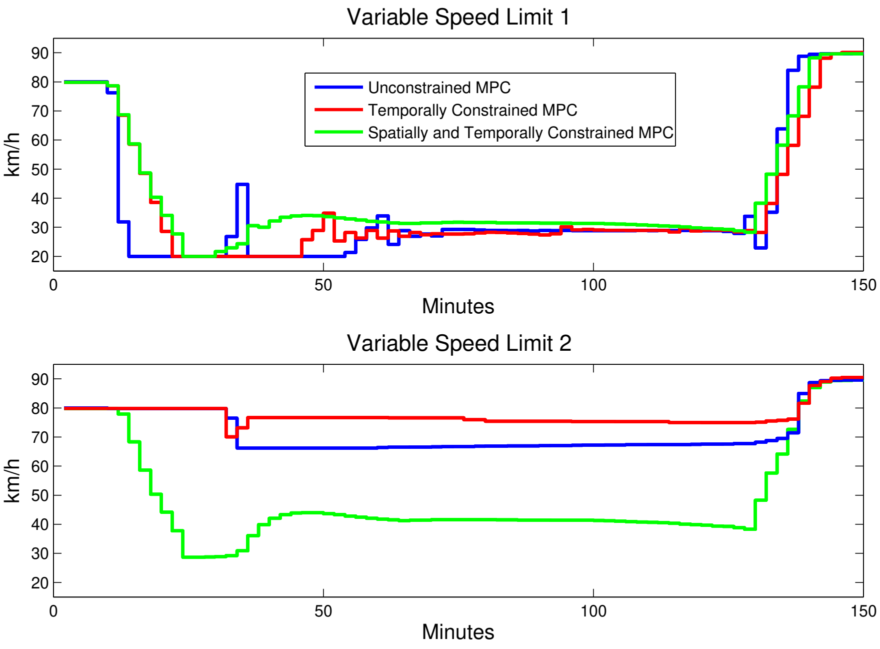

Fig. 8. VSL of segment 3 and 4 for continuous MPC and rounding discretization

In the numerical results on Table II can be seen how ST-MPC (the most realistic one) results on a TTS reduction that is the 36.7% lower than the TTS reduction corresponding to unconstrained MPC (12.47% versus 8.14%) concluding than the spatial and temporal constraints substantially decrease the potential controller performance. However, in real implementations, the lack of safety constraints may increase the actual TTS reduction due to sudden braking and accidents.

TABLE II

MPC PERFORMANCES

	TTS Reduction (%)
Uncontrolled System	0 %
Unconstrained MPC	12.87 %
Temporally constrained MPC (T-MPC)	10.48 %
Spatially and temporally constrained MPC (ST-MPC)	8.14 %

C. Rounding, Flooring and Ceiling

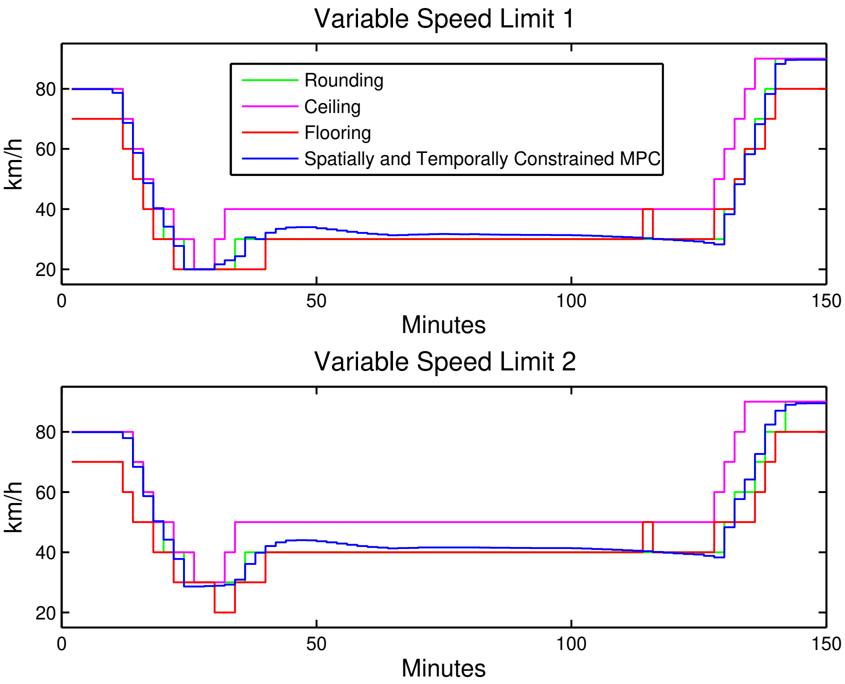

Fig. 9. Rounding, Ceiling, Flooring and continuous VSL (ST-MPC) of segment 3 and 4

The closed loop VSL discretization by rounding, flooring, and ceiling, and the continuous ST-MPC solution are shown in Fig. 9. It can be seen than different VSL choices during a few sample step during the beginning of the congestion can cause a different simulation during a long period of time (the ceiling VSL are 20 km/h and 30 km/h over the rounding and ceiling VSL during more than 90 minutes).

The numerical results of the rounding, ceiling, and flooring cases can be seen in Table III. In this simulation, the three options give a similar performance with a slight advantage for the rounding case. However, the TTS reduction for the rounding case is 46% lower than the TTS reduction corresponding to ST-MPC (8.14% versus 4.40%). This shows that the supposition done in many previous papers about the continuous implementation of the VSL can entail a large loss of performance for the controlled system for some networks and some traffic conditions.

TABLE III

MPC Performances

	TTS Reduction (%)
Spatially and temporally constrained MPC	8.14 %
Rounding discretization	4.40 %
Ceiling discretization	4.37 %
Flooring discretization	4.22 %

D. θ-Genetic and θ-Exhaustive Optimizations

The VSL discretization of ST-MPC by θ-Genetic and θ-Exhaustive optimization are shown in Fig.10. It can be seen than the solutions of both the genetic and the exhaustive optimizations keep around the continuous solution during the whole simulation. During the time that the continuous solution stays relatively constant, the discrete solutions approximate the ST-MPC switching between two discrete values in the same way as in a pulse-width modulation.

The numerical results can be seen in Table IV. It can be seen that values of θ higher than 10 do not bring much increase in the performance of the controlled system as can be seen in TTS reduction of the case with $\theta = 18$ and $\theta = 25$. More concretely, the θ-exhaustive optimization with $\theta = 10$ is only 1.6% worse than for $\theta = 25$. However, the computation

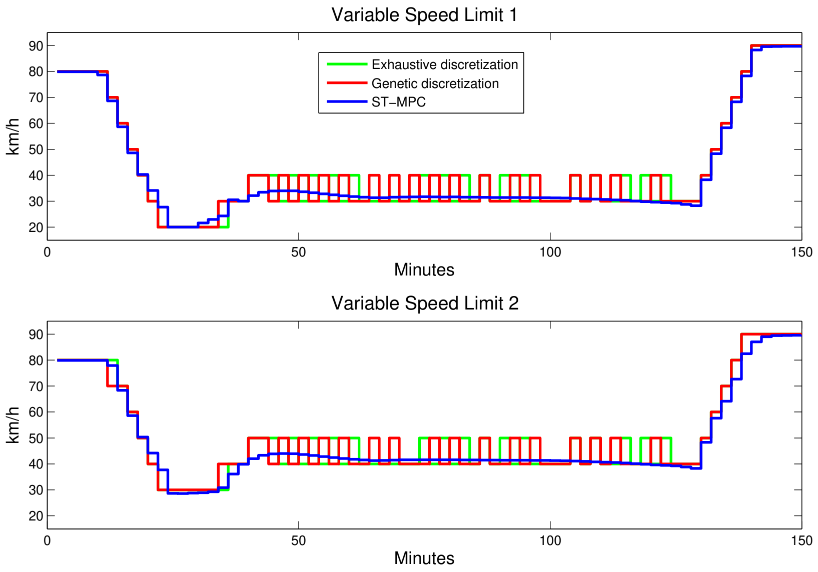

Fig. 10. Exhaustive and genetic optimizations

TABLE IV

θ-GENETIC AND θ-EXHAUSTIVE PERFORMANCES

	θ	TTS Reduction	Discrete optimization time
θ-Exhaustive optimization	10	7.22%	1.85 s
θ-Exhaustive optimization	14	7.24%	14.12 s
θ-Exhaustive optimization	18	7.25%	16.26 s
θ-Exhaustive optimization	25	7.34%	26.70 s
θ-Genetic optimization	10	6.98%	0.4 s

time is increased from $1.85s$ to $26.70s$. Therefore, we decided to use a value of $\theta = 10$ for the θ-Genetic optimization algorithm.

After a tuning process (equivalent to the one shown in the following subsection for the alternating genetic optimization), 20 individuals with 100 genes were chosen for the GA. A crossover parameter of 0.8 and a mutation parameter of 0.01 are used. It can be seen than the θ-Genetic optimization gives a solution that is close to the θ-Exhaustive one (6.98% versus 7.22%) but with a lower computation times (0.4 s versus 1.85 s).

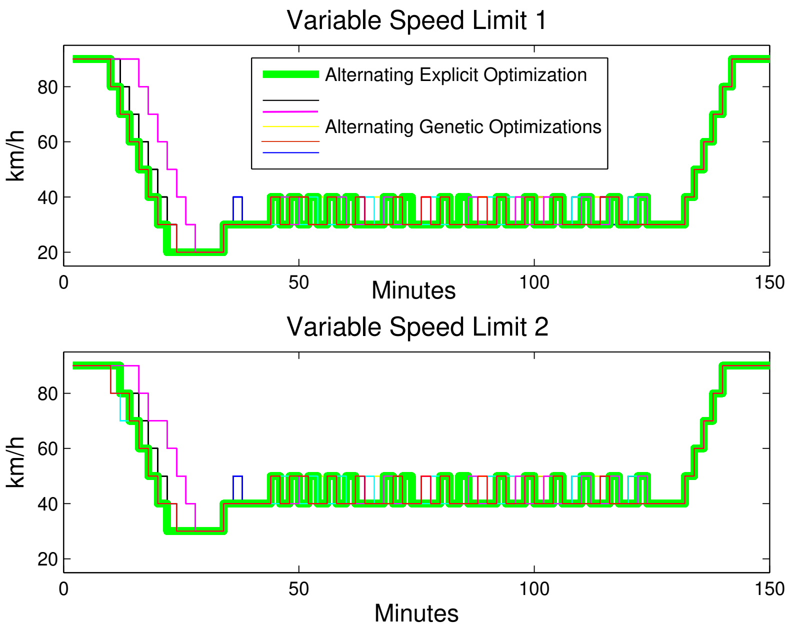

Fig. 11. Exhaustive optimization versus genetic optimizations

E. Alternating Optimization results

The VSL obtained by Alternating Exhaustive Optimization and six examples of the VSL obtained by using Alternating Genetic Optimization are shown on Fig.11. It can be seen than for the majority of the profiles obtained by the GA, the VSL stay very close to the exhaustive solutions.

TABLE V

TUNING OF ALTERNATING EXHAUSTIVE OPTIMIZATION

	Ramp Iterations	VSL Iterations	TTS Reduction (%)
Alternating Exhaustive Optimization	1	1	7.4229
Alternating Exhaustive Optimization	2	1	7.4252
Alternating Exhaustive Optimization	3	3	7.4252

The numerical results of the exhaustive optimization for different numbers of iterations are shown in Table V. It can be seen that the iterations almost do not increase the TTS reduction (7.4229% without iterations versus 7.4252% with three iterations for the ramp metering and the VSL). In fact, in this simulation the algorithm converges in one iteration of the ramp metering for 72 out the 75 sample times. Therefore, we decided to use only one iteration for the exhaustive and genetic optimizations (i.e. ramp metering optimization and VSL optimization will be run just one time each sample time). As the algorithm converges with three iterations for all the sample times and assuming that the ramp metering optimization is using enough initial points to avoid local minima, it can be said that 7.4252% is likely to be the maximum TTS reduction that can be achieved with the used horizons and cost function parameters. It will be difficult to implement the exhaustive algorithm in real time for large networks due the computation times required. However, in this case it is possible to compute the control inputs in real time. For the exhaustive optimization with just one iteration, the mean computation time for one sample step is 31.04 s with a maximum of 40.64 s which is less than the 2 minutes controller sample time. Using 3 iterations for the ramp metering and for the VSL, the mean computation time is 95.43 s with a maximum of 125.78 s.

TABLE VI

TUNING OF ALTERNATING GENETIC OPTIMIZATION

Ind	Gen	Muta	Cros	MTR(%)	SD-TR	Mean CT	SD-CT
20	100	0.01	0.8	6.99 %	0.29	6.14 s	1.71
20	100	0.001	0.8	5.81 %	1.30	7.22 s	1.81
40	200	0.01	0.8	7.37 %	0.14	12.55 s	2.65
40	200	0.001	0.8	7.13 %	0.38	9.85 s	1.93
60	100	0.01	0.6	7.27 %	0.178	8.45 s	1.01
30	100	0.01	0.6	7.11 %	0.36	6.26 s	1.91

Some of the numerical results obtained during the tuning of the Alternating Genetic Optimization are shown on Table VI. In the Table, Ind, Gen are the number of individuals and genes in the GA, Muta and Cros are the Mutation and Crossover parameter of the GA. MCT is the Mean Computation Time, and SDCT is the Standard Deviation of the Computational Time. In order to evaluate the performance of the genetic algorithm, the simulations have been run 10 times and the mean TTS reduction (MTR) and the standard deviation of this reduction (SDTR) have been used as performance measures.

The controllers show a good behavior for the majority of the parameters sets evaluated, usually with more than a 7% TTS reduction (close to the maximum of 7.43 %). Moreover, the standard deviation of the TTS reduction is relatively small showing a good performance even in the worst cases. It can also be seen that the trade-off between computation time and performance can be easily done by changing the number of individuals and genes.

VII. SUMMARY OF RESULTS AND CONCLUSIONS

A. *Summary of numerical results*

A brief summary of the numerical results and the computation times can be seen in Table VII. In the table, the row corresponding to Implementable shows whether the corresponding controller is implementable in terms of respect of the safety constraints and belonging to the discrete set of VSL. Maximum CT is the maximum computation time for all the sample times. It can be seen that the computation times for θ-Exhaustive and θ-Genetic optimizations are of the same order of magnitude as those of the continuous cases because the discrete optimization is faster than the continuous optimization. But the TTS associated to both θ-Exhaustive and θ-Genetic optimizations is significantly larger than for the rounding case (7.24 % and 7.22 % versus 4.40 %). Therefore, it can be concluded that if the algorithm is spending much computational load in order to computing a good solution for the continuous optimization, it is necessary to run a proper algorithm to obtain a good VSL signals. Otherwise, if we just use a rounding (or ceiling or flooring), we would be wasting the computational effort spent in the continuous optimization.

TABLE VII

CONTROLLERS SUMMARY

	Implementable	Mean CT	SD-CT	Max CT	MTR	SD-MTR
U-MPC	N	20.74 s	13.06	79.67 s	12.87%	0
T-MPC	N	22.39 s	10.09	49.34 s	10.48%	0
ST-MPC	N	30.02 s	13.67	88.76 s	8.14%	0
Rounding discretization	Y	31.39 s	16.34	78.90 s	4.4%	0
θ-Exhaustive optimization	Y	32.62 s	18.23	73.33 s	7.22%	0
θ-Genetic optimization	Y	31.91 s	17.11	64.21 s	6.98%	0.03
Alternating Exhaustive Optimization	Y	31.04 s	4.13	40.64 s	7.43%	0
Alternating Genetic Opt. (40/200/0.01/0.8)	Y	12.55 s	2.65	27.55 s	7.37%	0.14
Alternating Genetic Opt. (30/100/0.01/0.6)	Y	6.26 s	1.91	13.61 s	7.11%	0.36

From Table VII can be also concluded that both Alternating Optimizations have quite lower computation times than the other algorithms and a slightly greater TTS reduction. Especially important is the difference in the Max CT and in the standard deviation of these computation times (SDCT). This means that the computation times for the genetic and exhaustive optimization are more stable and, therefore, the inputs can be more likely computed during the available time.

Comparing the previously proposed solutions for real implementations (rounding, ceiling, flooring) with the genetic optimization it can be seen that the loss of performance corresponding to the rounding is much higher that the corresponding to the suboptimality of the genetic optimization.

B. Observed advantages of genetic optimization for freeway traffic control

- In many cases in freeway traffic control with METANET, continuous MPC based on an SQP optimization algorithm shows a poor performance if a wrong set of initial points is used for the optimization. For example, in this paper it was used an evaluation procedure (with 252 points conveniently chosen) for the MPC constrained cases in order to choose a proper initial point for the optimization in a reasonable time. In general, it would be necessary to run many optimizations with different initial points. To have to run multiple optimizations makes it impossible to compute the control inputs in real time. The genetic optimization is able to compute a reasonably good solution faster than the continuous optimization, allowing obtaining in real time a suboptimal solution in cases where the continuous optimization is unable due to the high number of initial points required.

- For the Alternating Optimizations the computation times for the different sample times are lower and with less variation than for typical continuous MPC approaches as can be seen in Table VII (the standard deviation and the mean of the computational times are much lower than for the continuous cases). This allows making the most of the time available ensuring that there will be a suboptimal implementable solution at the end of the time step.

- The genetic and exhaustive optimization substantially improve the performance of the rounding discretization (7.24 % and 6.98 % versus 4.20 %), which is the best implementable solution without running a discrete optimization. Moreover, these algorithms can be run after the continuous optimization in only 1.86 s and 0.4 s, respectively. Both solutions are close to the best implementable solution (7.43%).

C. *Further remarks and future work*

- Using genetic algorithms it is impossible to ensure reaching the global optimum within a limited sampling time. Moreover, stability or other robust properties can be analyzed just numerically. However, using continuous optimization it is also impossible in practice to prove stability or to ensure global optimality when solving highly non-linear non-convex optimization problems.

- GA perform in a different way any time they are executed so the proposed VSL profiles will randomly show slight changes for one simulation to another. However, in this benchmark the observed standard deviation of the TTS reduction is low.

- For large networks, where a distributed algorithm is required [8], the iterations between the continuous ramp metering problem and the discrete VSL problem may use information from neighbors controllers (corresponding to nearby regions of the freeway) in order to run a distributed optimization which tries to approach the global minimum using a equivalent procedure that the one used in [7] for continuous controllers.

- As the methods we propose are independent of the traffic model used, it would be interesting to apply them using other macroscopic traffic models which are capable of including the effect of VSL in their formulation (like some versions of the Cell Transmission Model CTM [16]).

D. *Conclusions*

This paper has proposed hybrid Model Predictive Control (MPC) approaches for freeway traffic control considering Variable Speed Limits (VSL) as discrete variables as in current real world implementations.

Firstly, this paper has analyzed the effect of converting the continuous unconstrained VSL signal to an implementable VSL signal (i.e. a discrete value respecting the safety constraints). In the simulated network, the discrete constrained performance is substantially reduced with respect to the ideal case. Therefore, it can be concluded that the assumption about the continuous implementation of the VSL entails in some cases a large loss of performance for the controlled system. Since the majority of the literature makes this supposition, we have proposed two kind of methods to reduce part of this loss of performance.

The first class of proposed methods discretize the VSL signals solving a discrete optimization problem with a search space limited to a θ distance from the continuous MPC solution. Using an exhaustive evaluation of all the feasible discrete solutions, it is possible to substantially improve the solution. In the network simulated in this paper, the computation can be done in a short time compared to the continuous MPC. In cases where this method entails unacceptable computation times, it is proposed to relax the exhaustive search using a genetic algorithm. In the analyzed simulation, the θ-Genetic optimization is able to closely approximate the behavior of the θ-Exhaustive optimization but with a reduced computation time.

Finally, we have proposed an algorithm that computes the discrete VSL directly (i.e. without using the solution provided by the continuous MPC optimization). The mixed-integer optimization is solved by computing the ramp metering and the VSL iteratively and separately in order to decompose the problem in one continuous optimization problem and one discrete optimization problem. The continuous optimization can be solved by using, for instance, common SQP algorithm. For the discrete optimization, an exhaustive algorithm can be used and, again, a GA is proposed in order to solve the optimization in a reasonable time.

The results show that, for the given case study, the exhaustive optimization can be executed in real time substantially improving the performance of the rounding discretization, which is the best implementable solution without running a discrete optimization. In case where the exhaustive solution cannot be found in real time, the alternating genetic optimization almost reaches the behavior of the exhaustive one. The alternating genetic optimization does not use the solution of the continuous MPC problem and so reduces the computation times. Moreover, the trade-off between simulation speed and accuracy can be easily adapted to the available time making the algorithm very useful for the application in practical cases, especially if it is necessary to use a distributed scheme due to the size of the network.

REFERENCES

[1] A. Hegyi. *Model Predictive Control for Integrating Traffic Control Measures*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2004.

[2] A. Muralidharan and R. Horowitz. Optimal control of freeway networks based on the link node cell transmission model. In *Proceedings of the American Control Conference (ACC)*, pages 5769–5774, 2012.

[3] A. Hegyi and S.P. Hoogendoorn. Dynamic speed limit control to resolve shock waves on freeways - field test results of the specialist algorithm. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 519–524, Madeira Island, Portugal, 2010.

[4] E. van den Hoogen and S. Smulders. Control by variable speed signs: results of the dutch experiment. In *Proceedings of the 7th International Conference on Road Traffic Monitoring and Control*, pages 145–149, 1994.

[5] M. Papageorgiou, J.M. Blosseville, and H. Haj-Salemn. Modelling and real-time control of traffic flow on the southern part of boulevard priphrique in paris: Part ii: coordinated on-ramp metering. *Transportation Research Part A*, 24(5):361–370, 1990.

[6] E.F. Camacho and C.A. Bordons. *Model Predictive Control*. Springer-Verlag, London, U.K., 2004.

[7] J.R.D. Frejo and E.F. Camacho. Feasible cooperation based model predictive control for freeway traffic systems. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5965–5970, Orlando, FL, USA, 2011.

[8] J.R.D. Frejo and E. F. Camacho. Global versus local MPC algorithms in freeway traffic control with ramp metering and variable speed limits. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1556–1565, 2012.

[9] S.K. Zegeye, B. De Schutter, J. Hellendoorn, E.A. Breunesse, and A. Hegyi. A predictive traffic controller for sustainable mobility using parameterized control policies. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1420–1429, 2012.

[10] G. Gomes and R. Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006.

[11] A. Hegyi, B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):102–112, 2005.

[12] R. Castelan-Carlson, I. Papamichail, and M. Papageorgiou. Local feedback-based mainstream traffic flow control on motorways using variable speed limits. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1261–1276, 2011.

[13] A. Hegyi, S.P. Hoogendoorn, M. Schreuder, H. Stoelhorst, and F. Viti. SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 827–832, Beijing, China, 2008.

[14] C. Canudas de Wit. Best-effort highway traffic congestion control via variable speed limits. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5959–5964,

Orlando, FL, USA, 2011.

[15] M. Papageorgiou, I. Papamichail, A. Messmer, and Y. Wang. Traffic simulation with METANET. In J. Barceló and F.S. Hillier, editors, *Fundamentals of Traffic Simulation*, volume 145 of *International Series in Operations Research and Management Science*, pages 399–430. Springer New York, 2010.

[16] C.F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

[17] M. Papageorgiou, E. Kosmatopoulos, and I. Papamichail. Effects of variable speed limits on motorway traffic flow. *Transportation Research Record*, 2047(1):37–48, 2008.

[18] A. Hegyi, B. De Schutter, and H. Hellendoorn. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, 13(3):185–209, 2005.

[19] O Slupphaug, J Vada, and B Foss. MPC in systems with continuous and discrete control inputs. In *Proceedings of the 1997 American Control Conference*, June 1997.

[20] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–427, 1999.

[21] A. Bemporad, W. Heemels, and B. De Schutter. On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control*, 47(5):863–869, 2002.

[22] C.A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Topics in Chemical Engineering. Oxford University Press, USA, 1995.

[23] A. Ingimundarson, C. Ocampo-Martínez, and A. Bemporad. Suboptimal model predictive control of hybrid systems based on mode-switching constraints. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, pages 5264–5269, New Orleans, LA, USA, 2007.

[24] K.F. Man, K.S. Tang, and S. Kwong. *Genetic Algorithms: Concepts and Designs*. Springer-Verlag, Secaucus, NJ, USA, 2nd edition, 1999.

[25] H. Sarimveis and G. Bafas. Fuzzy model predictive control of non-linear processes using genetic algorithms. *Fuzzy Sets and Systems*, 139(1):59–80, 2003.

[26] J.H. van der Lee, W.Y. Svrcek, and B.R. Young. A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making. *ISA Transactions*, 47(1):53–59, 2008.

[27] A. Núnez, D. Sáez, S. Oblak, and I. Škrjanc. Fuzzy-model-based hybrid predictive control. *ISA Transactions*, 48(1):24–31, 2009.

[28] Z. Michalewicz. Do not kill unfeasible individuals. In Dabrowski, Michalewicz, and Ras, editors, *Proceedings of the Fourth Intelligent Information Systems Workshop (IIS'95)*, pages 110–123, Augustow, Poland, 5–9 June 1995.

[29] T. Back, D. Fogel, and Z. Michalewics, editors. *Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK, 2000.

[30] C. Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.

[31] S. Kimbrough, G. Koehler, M. Lu, and D. Wood. On a feasible-infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *European Journal of Operational Research*, 190(2-16):310–327, 2008.

[32] Z. Michalewicz and G. Nazhiyath. Genocop iii: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *IEEE International Conference on Evolutionary Computation*, volume 2, pages 647–651, 1995.