International Journal of
Mathematical Modelling & Computations
Vol. 13, No. 01, winter 2023, 1- 21
DOI:10.30495/ijm2c.2023.1951076.1244

Approximate solution of general mp-MILP problems and its application in urban traffic networks

M. Mahmoudi [a,*], A. Heidari [a] and A.karimpour[b]

[a] *Department of Mathematics, Payame Noor University (PNU), P. O. BOX 19395-3697, Tehran, Iran*

[b] *Department of Electrical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, P. O. BOX 91775-1111 Mashhad, Iran*

Abstract. The multi-parametric programming (mp-P) is designed to minimize the number of unnecessary calculations to obtain the optimal solution under uncertainty, and since we widely encounter to that kind of problem in mathematical models, its importance is increased. Although mp-P under uncertainty in objective function coefficients (OFC) and right-hand sides of constraints (RHS) has been highly considered and numerous methods have been proposed to solve them so far, uncertainty in the coefficient matrix (i.e., left-hand side (LHS) uncertainty) has been less considered. In this work, a new method for solving multi-parametric mixed integer linear programming (mp-MILP) problems under simultaneous uncertainty OFC, RHS, and LHS is presented. The method consists of two stages which in the first step, using tightening McCormick relaxation, the boundaries of the bilinear terms in the original mp-MILP problem are improved, the approximate model of the problem is obtained based on the improved boundaries of the first stage, and finally an algorithm is presented to solve these kinds of problems. The efficiency of the proposed algorithm is investigated via different examples and the amount of required calculations for solving the problem in different partitioning factors is compared. Also, model predictive control (MPC) using mp-P is designed for an example of urban traffic network to examine the practical application of the proposed algorithm.

Received: 31 January 2022, Revised: 01 march 2023, Accepted: 01 march 2023.

Keywords: Multi-parametric programming (mp-P), Optimization under uncertainty, Mixed-integer programming (MIP), Explicit model predictive control (EMPC), Urban traffic network.

AMS Subject Classification: 90C11; 90C31; 90C99.

Index to information contained in this paper

*Corresponding author Email: mrym.mahmoudi@gmail.com

1. Introduction

The principal aim of programming is to control future events in accordance with the objectives so that it is more prepared to deal with upcoming changes. Uncertainty exists widely in the models presented for physical phenomena, so it is important to investigate the changes in the optimal solution obtained after solving the model. In multi-parametric programming (mp-P), an optimization problem is solved for a range and as a function of multiple parameters. In recent decades, mp-P has received a lot of attention. Applications of that can be seen in model predictive control (MPC) [26, 30–32, 34] scheduling under uncertainty [23, 33], and bi-level and dynamic programming [4–6, 13].

In spite of good advances for certain classes of mp-P problems such as multi-parametric linear programming (mp-LP), mp-MILP, multi-parametric mixed integer programming (mp-MIP), and multi-parametric quadratic programming (mp-QP), with OFC and RHS uncertainty, or a combination of both, models with an LHS uncertainty have been received less attention. The problem facing these problems is that owing to the presence of the parameter on the left-hand side of the constraints, a bilinear term is created which is followed by the computational complexity of the resulting problem.

In [28] two algorithms for solving mp-MILPs are presented in general for when we have a single parameter uncertainty at the same time, the first is based on the branch and bound algorithm and solving parametrically an LP in each single node, and the second divides the problem into MILP, mp-LP and MINLP sub-problems. In both algorithms presented in [28] to solve linear parametric sub-problems, the algorithm presented in [11] is used and their algorithm can be used even in the specific single parameter studied for problems with limited variable numbers. In [22], a method for solving a small-sized mp-MILP problems with general uncertainty based on the branch and bound algorithm and using the optimality conditions for the standard linear optimization problem is presented. In the circumstances that the uncertainty appears on the left side, some researchers apply projection algorithms [3] based on convex hulls for a full explanation of the critical region, and the type of projection algorithms that they use have an influence on the results that they acquired. In [37] an approximation algorithm for global solutions of general mp-MILP problems is presented. They transform the LHS uncertainty into the RHS uncertainty by using McCormick relaxation, as well as employ the logarithmic partitioning scheme for parameter space to cut down the number of decision variables. Then by using that approximation and applying the two-step method [36], they have presented a general piecewise affine relaxation for this kind of problems which as the number of the Partitioning number increases, the number of subproblems that must be solved in linear and logarithmic states increases linearly and logarithmically respectively. An algorithm for an LHS single-parameter uncertainty in the LP problem is considered in [20]. The authors had developed an algorithm to find optimal values for the entire range of parameters by using a two-step iterative method. At first, by applying the Flavell-Salkin approximate approach [14], the optimality condition equations is solved and then the location of break-points is found. In the second step, the rigorous Woodbury equation [17] is used to check the accuracy of the results and they will be up-to-date if it is necessary. In [10], the authors suggest an analytical solution to solve mp-MILPs under global uncertainty based on the principles of symbolic computation and semi-algebraic geometry. They solve the Karush Kuhn Tucker (KKT) system of the original problem by using Groebner Bases theory [35] within a symbolic computational environment and obtain candidate solutions, and then through using

the optimality and feasibility conditions, the global parametric optimal solutions of the problem is achieved. One of the important features of that algorithm is the precise computation of non-convex critical regions, but when the number of constraints and variables increases the number of initial candidate solutions grows.
As stated above, researches for problems with general uncertainty can be divided into three categories, based on the relaxation of bilinear terms using such as McCormick relaxation, Exact algorithm in single-parametric mode, and the use of KKT conditions and symbolic environments to obtain accurate answers. The main problem of all three mentioned methods is the limitation of its application in the number of high variables because of the high computational volume, which shows the importance of using a method to reduce the amount of calculations more than before. To solve this problem, in this work, firstly by using the McCormick relaxation bilinear terms, resulting from the product of decision variable and the uncertainty parameter, convert to the linear constraints and their bounds will be improved. In the next step, a piecewise affine approximation of bilinear terms is presented by applying the method where is introduced in [15] and bounds which are obtained from the first step are improved. So this approximate model is an RHS-mp-MILP problem. Then, the decomposition algorithm is implemented to solve these problems [12] and using the available Matlab toolbox to solve mp-LPs, YALMIP [24], an algorithm for mp-MILPs is presented in general. It is important to note that the toolboxes provided for solving mp-MILP problems are only applicable if we have uncertainty on the right-hand side, and none of the toolboxes presented so far is able to solve mp-MILP problems with simultaneous uncertainty, especially the simultaneous uncertainty of LHS and OFC or LHS and RHS. The efficiency of the proposed new algorithm for solving a general mp-MILP in order to reduce the amount of calculations has been compared with another algorithm for solving these problems presented in [37] and critical regions and the optimal solution obtained from the new algorithm with critical regions and the optimal solution presented in [10] are reported.
In addition, an example of urban traffic control is considered to demonstrate a practical application of the proposed algorithm. As one of the most powerful and widely used control technologies, MPC has been widely used in Traffic Signal Control (TSC) and significant results have been obtained. MPC methods have been widely used in urban traffic networks [2], [16], [1] but they are often designed as a deterministic framework, while uncertainty mostly existed in real traffic networks. Hence uncertainty (eg traffic demand, random disturbances) should also be considered in traffic modeling.
Due to the complexity of the urban traffic system, the MPC approaches mentioned for TSC try to balance the accuracy of the model with the calculation time for its implementation. However, the biggest challenge for implementing MPC is the complexity of its online optimization computing. To resolve this, some methods have been used to reduce online computing, one of which is to solve the problem offline with mp-P. Recently, optimization for control signal splitting for large-scale traffic networks based on the store-and-forward model using explicit predictive model-based control (EMPC) has been introduced [26]. Their results show that by converting online computing to offline by mp-P, the efficiency of the designed controller is significantly increased and makes it available in real urban traffic networks. Although their method reduces the complexity of computing compared to the standard MPC, it is designed in a deterministic framework. In addition, turning times and travel times on the roads between intersections are ignored. On the other hand, the MPC approach proposed in [21] is a successful approach to simultaneously optimize green times duration and turning fractions at intersections to

minimize the number of vehicles in the controlled traffic networks, which also takes into account travel time.

The model proposed in [21] uses linear constraint to simulate real traffic behavior that is also very efficient in MPC. According to the knowledge of the authors, EMPC does not design for it yet. In this work we design EMPC for urban traffic networks using the new proposed algorithm and the model presented in [21]. In the mathematical linear models proposed in [21], the complexity of the optimization problem created by MPC is reduced by considering the traffic flow as a kwon and constant parameter. If we consider the traffic flow as a parameter, ans design an EMPC for the new model, we are faced with a mp-P that uncertainty appears on the LHS and RHS of the optimization problem. The resulttiong optimization problem can be solved by the new proposed algorithm. By this method, we consider the uncertainty for traffic demand and optimal control is designed offline which avoids solving an optimization problem online at any time step. As a result, its practical application in real urban traffic networks is possible because it significantly reduces the computational complexity.

The rest of the paper is organized as follow: First of all, in section two we are going to improve the bounds of bilinear variables. Secondly, in section three an approximate model will be created by using the improved bounds from the section two. Finally, in section four we will show the applicability of the proposed method with some numerical examples that clarify the computational results and will compare them with another methods.

2. Algorithm to improve the bounds of bilinear terms in mp-MILP

McCormick relaxation is an efficient technique for the linearization of nonlinear terms [27]. The tightness of bilinear terms are investigated in [15]. In this section, mp-MILPs with global uncertainty is defined, and tightening piecewise McCormick relaxation is applied to obtain an improved bound for variables that appear in bilinear terms. In general, an mp-MILP problem is

$$\begin{cases} z(\theta) = min_{x,y}((c+H\theta)^T x + (d+R\theta)^T y), \\ subject\ to: \quad A(\theta)x + E(\theta)y \leqslant b + F\theta, \\ x \in X = \{x \in \mathbb{R}^{n_x} | x_j^{min} \leqslant x_j \leqslant x_j^{max}, j = 1, \dots, n_x\}, \quad y \in \{0,1\}^p, \\ \theta \in \Theta = \{\theta \in \mathbb{R}^{n_\theta} | \theta_l^{min} \leqslant \theta_l \leqslant \theta_l^{max}, l = 1, \dots, n_\theta\}, \end{cases} \tag{1}$$

The notation used in the mp-MILP problem and its approximation process is defined in Table 1.

Let the coefficients matrix corresponding to the constraints is linearly dependent on the parameter θ and is defined as follows

$$\begin{aligned} A(\theta) &= A^c + \sum_{l=1}^{n_\theta} A_l^p \theta_l, \quad A^c = [a_{ij}]_{m\times n_x}, \quad A_l^p = [a_{ij}^l]_{m\times n_x}, \\ E(\theta) &= E^c + \sum_{l=1}^{n_\theta} E_l^p \theta_l, \quad E^c = [e_{ik}]_{m\times p}, \quad E_l^p = [e_{ik}^l]_{m\times p}. \end{aligned} \tag{2}$$

The objective of mp-MLIP problem is to determine the $z(\theta)$ as a function of θ in each crossponding critical regions (CR). If the uncertainty in the technology matrix is ignored, $A(\theta) = A^c, E(\theta) = E^c$, problem (1) have an OFC and RHS uncertainty and numerous methods have been suggested to solve these kind of optimization problems up to now. But very little work exists on the solution of

Table 1. List of symbols used in this article

symbol	remark
c	Coefficient vector of continuous variable x in objective function, $c \in \mathbb{R}^{n_x}$.
d	Coefficient vector of binary variable y in objective function, $d \in \mathbb{R}^{p}$.
$A(\theta)$	Coefficients matrix of x which linear based on θ, $A(\theta) \in \mathbb{R}^{m\times n_x}$.
$E(\theta)$	Coefficients matrix of y which linear based on θ, $E(\theta) \in \mathbb{R}^{m\times p}$.
F	Coefficients matrix of uncertainty parameter θ in right-hand side of constraints, $F \in \mathbb{R}^{m\times n_\theta}$.
b	Right-hand side of constraints, $b \in \mathbb{R}^{m}$.
A^c	Fixed part of $A(\theta)$
E^c	Fixed part of $E(\theta)$
A^p_l	Coefficients matrix of uncertainty parameter in $A(\theta)$.
E^p_l	Coefficients matrix of uncertainty parameter in $E(\theta)$.
H	Coefficients matrix of θx in objective function, $H \in \mathbb{R}^{n_x\times n_\theta}$.
R	Coefficients matrix of θy in objective function, $R \in \mathbb{R}^{p\times n_\theta}$.
j	Component counters of x
k	Component counters of y
l	Component counters of θ
θ_l^{min} and x_j^{min}	Lower bounds of θ_l and x_j
θ_l^{max} and x_j^{max}	Upper bounds of θ_l and x_j
θ_{ln}^{min} and θ_{ln}^{max}	Lower nad upper bounds of θ_l in n-th partition
x_{jln}^{min} x_{jln}^{max}	Lower nad upper bounds of x_j when θ_l is in the n-th partition
w_{jl}	The product of continuous variable x_j and uncertainty parameter θ_l
v_{kl}	The product of binary variable y_k and uncertainty parameter θ_l
γ_n^l	Binary variable to display being active θ_l in n-th partition
λ_q^j	Binary variable to display being active x_j in q-th partition
δ_{lq}^j	Continuous variable corresponding to variable x_j and parameter θ_l in q-th partition
N_l	Partitioning number of parameter θ_l
M_j	Partitioning number of variable x_j
β^j	The increment of x_j
BL	A set of indexes (j,l) in bilinear terms
DL	A set of indexes (k,l) in bilinear terms
$X \subseteq \mathbb{R}^{n_x}$	Definition region of continuous variables $x \in X$
$\Theta \subseteq \mathbb{R}^{n_\theta}$	Definition region of uncertainty parameters $\theta \in \Theta$

problem (1) which contain simultaneous uncertainty, beacuse of the non-convexity of the problem (the existence of the sentences θx and θy), and high computational complexity. It is of special importance to fix non-convexity of $A(\theta)$ and $E(\theta)$ that one of them is linearization. Bilinear terms of (1) regarding (2) is made up of θx and θy that McCormick relaxations [27] is used to linearize them. For example, to relax θx by taking the new variable w, inequality corresponding to the variable and parameter that have created the bilinear term, $w = \theta x$, is

$$\begin{aligned} w &\geqslant x\theta^{min} + x^{min}\theta - x^{min}\theta^{min},\\ w &\geqslant x\theta^{max} + x^{max}\theta - x^{max}\theta^{max},\\ w &\leqslant x\theta^{min} + x^{max}\theta - x^{max}\theta^{min},\\ w &\leqslant x\theta^{max} + x^{min}\theta - x^{min}\theta^{max}, \end{aligned} \tag{3}$$

The relaxation for θy is similarly definable. Concerning 3, as we select tighter upper and lower bounds we will obtain the higher quality of the relaxation, and thus the search space is shrunk.

In [15], the authors prove that the quality of the McCormick relaxation is affected by the upper and lower bounds of each variables that appears in the bilinear terms. Accordingly, a new method based on improving the boundaries of bilinear variables and thus improving the efficiency of McCormic relaxation and its use in solving multi-parametric programming problems is presented in this paper.
To shrink the search space in the feasible region of the relaxed problem, we need to partition the domain of the problem. At this point of approximation it is time to consider partitioning of the uncertainty parameter. Partitioning can be made uniformly and/or nonuniformly [9].
Have a look at (1) and the uncertainty parameter θ_l, $\theta_l^{min} \leqslant \theta_l \leqslant \theta_l^{max}$ also assume that BL is the set of all (j, l), and DL is the set of all (k, l) in bilinear terms, where j, k and l is the index of variables x, y and parameter θ respectively. Let the N_l as the partition number of θ_l divide Θ into N_l equal sub-intervals

$$\theta_l^{min} + \frac{(\theta_l^{max} - \theta_l^{min})(n-1)}{N_l} \leqslant \theta_l \leqslant \theta_l^{min} + \frac{(\theta_l^{max} - \theta_l^{min})n}{N_l}, \quad n = 1, 2, \ldots, N_l (4)$$

Let θ_{ln}^{min} and θ_{ln}^{max} are respectively the lower and upper bounds of θ_l, within the n^{th} partition, $n = 1, 2, \ldots, N_l$. Now we suppose $w_{jl} = x_j\theta_l$ and define $v_{kl} = y_k\theta_l$ as a new variable for the bilinear terms, which are produced from the product of binary variable and uncertainty parameter, so similar to (3) four corresponding constraints will be appear for this.
In the following, the bounds of the above-mentioned McCormick relaxation of (1) will improve by executing a maximization problem and a minimization problem, that this idea is taken from [9].

First, select a parameter θ_{l^*}, $l^* \in \{1, 2, , \ldots, n_\theta\}$ then the variable x_{j^*} such that $\{j^*|(j^*, l^*) \in BL\}$, and a partition $n^* \in \{1, 2, \ldots, N_{l^*}\}$. For that partition, bounds of θ_{l^*} are obtained from (4), and for the other variables, the original bounds of the problem are taken into account. Also assuming that the uncertainty parameter is a decision variable, the problem (1) by considering (2) is a nonlinear mixed-integer programming problem (NMILP) whose solution by helping a nonlinear solver can be an upper bound for the objective function that called it z'. Maximization and a minimization problem is constructed based on the given assumptions

$$\begin{aligned}
& x_{j^*l^*n^*}^{min} = \min_{x_j,\theta_l,y_k,w_{jl},v_{kl}} x_{j^*}, \quad or \quad (x_{j^*l^*n^*}^{max} = \max_{x_j,\theta_l,y_k,w_{jl},v_{kl}} x_{j^*}) \\
& subject\ to: \\
& \sum_{j=1}^{n_x}(c_j + \sum_{l=1}^{n_\theta} h_{jl}\theta_l)x_j + \sum_{k=1}^{p}(d_k + \sum_{l=1}^{n_\theta} r_{kl}\theta_l)y_k \leqslant z' \\
& \sum_{i=1}^{m}\sum_{j=1}^{n_x} a_{ij}x_j + \sum_{i=1}^{m}\sum_{k=1}^{p} e_{ik}y_k + \sum_{i=1}^{m}\sum_{l=1}^{n_\theta}\sum_{j=1}^{n_x} a_{ij}^l w_{jl} + \\
& \quad \sum_{i=1}^{m}\sum_{l=1}^{n_\theta}\sum_{k=1}^{p} e_{ik}^l v_{kl} \leqslant \sum_{i=1}^{m} b_i + \sum_{i=1}^{m}\sum_{l=1}^{n_\theta} f_{il}\theta_l \\
& \left.\begin{aligned}
w_{jl} &\geqslant x_j\theta_l^{min} + x_j^{min}\theta_l - x_j^{min}\theta_l^{min} \\
w_{jl} &\geqslant x_j\theta_l^{max} + x_j^{max}\theta_l - x_j^{max}\theta_l^{max} \\
w_{jl} &\leqslant x_j\theta_l^{min} + x_j^{max}\theta_l - x_j^{max}\theta_l^{min} \\
w_{jl} &\leqslant x_j\theta_l^{max} + x_j^{min}\theta_l - x_j^{min}\theta_l^{max}
\end{aligned}\right\} \forall (j, l) \in BL
\end{aligned}$$

$$\begin{array}{l}
\left.\begin{array}{l}
v_{kl} \geqslant y_k\theta_l^{min} + y_k^{min}\theta_l - y_k^{min}\theta_l^{min} \\
v_{kl} \geqslant y_k\theta_l^{max} + y_k^{max}\theta_l - y_k^{max}\theta_l^{max} \\
v_{kl} \leqslant y_k\theta_l^{min} + y_k^{max}\theta_l - y_k^{max}\theta_l^{min} \\
v_{kl} \leqslant y_k\theta_l^{max} + y_k^{min}\theta_l - y_k^{min}\theta_l^{max}
\end{array}\right\} \forall(k,l) \in DL \\
\left.\begin{array}{l}
w_{jl}\ \forall\{l|(j,l) \in BL\} \\
v_{kl}\ \ \forall\{l|(k,l) \in DL\}
\end{array}\right\} j \in \{1,2,\dots,n_x\}, k \in \{1,2,\dots p\} \\
\theta_{l^*n^*}^{min} \leqslant \theta_{l^*} \leqslant \theta_{l^*n^*}^{max} \\
x \in X = \{x \in \mathbb{R}^{n_x} | x_k^{min} \leqslant x_k \leqslant x_k^{max}, k = 1,\dots,n_x\} \\
\theta \in \Theta = \{\theta \in \mathbb{R}^{n_\theta} | \theta_l^{min} \leqslant \theta_l \leqslant \theta_l^{max}, l = 1,\dots,n_\theta, l \neq l^*\}.
\end{array} \qquad (5)$$

So tighter upper and lower bounds for x_j will be achieved if the problem (5) has a solution, where θ_{l^*} lies in the n^{th} partition. If (5) is infeasible, partition n^* which corresponds to θ_{l^*} is deleted and therefore smaller problem size for the latter is introduced. Replacing bounds x_j^{min}, x_j^{max} for x_j with improved bounds $x_{jln}^{min}, x_{jln}^{max}$, leads to a tighter relaxation. Now the obtained bounds is replaced with the original bounds of the problem and will be used to approximate the problem (1) in the next section.

It is important to note that by using this technique for bounds tightening we require the solution of multiple MILP problems, and the number of MILP subproblems at this stage increases linearly via the number of partitions. In other words, when the proposed method is applied to general mp-MILP problems, the size of MILPs seems to be a computational time issue. But our examples in the next section show, by increasing the partition, the number of RHS-mp-MILP to be solved is considerably reduced and improved bounds would help us to examine the critical region of the problem more effectively.

3. Linear approximate model of a general mp-MILP problem

In this section we are going to elucidate an approximation of mp-MILP based on [15] and the new bounds obtained from the previous section. By applying this technique, we approximate bilinear terms. Therefore the coefficient matrix in terms of decision variables is linearized and finally a coefficients matrix of the uncertainty parameter will be obtained on the RHS. In other words, the approximation of the original problem is finally an RHS- mp-MILP. Tighter bounds that are used to relax mp-MILP (1), transform it into an approximate model that computational requirements will be significantly reduced. But the number of sub-problems increases as the number of partitions increases, therefore one should set the value of the partitioning number in a trade-off between accuracy and complexity. To do this, problem (1) is transformed into an mp-MILP problem with LHS- and RHS-uncertainty by introducing an auxiliary variable μ

$$\begin{cases}
z(\theta) = min_{x,y,\mu}\mu \\
subject\ to \\
A(\theta)x + E(\theta)y \leqslant b + F\theta \\
(c + H\theta)^T x + (d + R\theta)^T y \leqslant \mu \\
x \in X = \{x \in \mathbb{R}^{n_x} | x_j^{min} \leqslant x_j \leqslant x_j^{max}, j = 1,\dots,n_x\} \\
y \in \{0,1\}^p, \quad \mu \in \mathbb{R} \\
\theta \in \Theta = \{\theta \in \mathcal{R}^{n_\theta} | \theta_l^{min} \leqslant \theta_l \leqslant \theta_l^{max}, \quad l = 1,\dots,n_\theta\}
\end{cases} \qquad (6)$$

and, then the upper and lower bounds of the variable x_j were improved in the previous section are replaced with bounds of the problem (6). In the following,

x_j^{min} and x_j^{max} are the solution of (5). The feasible region of x_j is divide into M_j equal intervals. It means the length of each interval is

$$\beta^j = \frac{x_j^{max} - x_j^{min}}{M_j}, \quad j = 1, 2, \ldots n_x. \tag{7}$$

To realize that a partition is whether active or inactive, we require a binary variable λ_q^j corresponding the variable x_j. When x_j lies in the q-th subinterval its value is one, otherwise zero. Since variable x_j always lies in one of the M_j subintervals it is obvious that

$$\sum_{q=1}^{M_j} \lambda_q^j = 1. \tag{8}$$

To model the domain of the variable x_j by helping the λ_q^j, we have [15]

$$x_j^{min} + \beta^j \sum_{q=1}^{M_j} (q-1)\lambda_q^j \leqslant x_j \leqslant x_j^{min} + \beta^j \sum_{q=1}^{M_j} q\lambda_q^j, \quad j = 1, 2, \ldots n_x. \tag{9}$$

It is also mandatory to define a set of continuous variables $\delta_{l,q}^j$ so that the following conditions are hold [15]

$$\begin{aligned} \theta_l = \theta_l^{min} + \sum_{q=1}^{M_j} \delta_{l,q}^j, \quad & l = 1, \ldots, n_\theta, j = 1, \ldots, n_x, \\ 0 \leqslant \delta_{l,q}^j \leqslant (\theta_l^{min} - \theta_l^{max})\lambda_q^j \quad & l = 1, \ldots, n_\theta, j = 1, \ldots, n_x, q = 1, \ldots, M_j. \end{aligned} \tag{10}$$

The equation (10) is an estimate for the deviation of variable θ_l from its lower bound whenever the variable x_j lies in the q^{th} subinterval. Now, the nonlinear terms of (1) with regards to (2) are estimated by employing McCormick envelopes [27]. The relations (9), (10), and the definition of McCormick envelopes result in

$$\begin{aligned} x_j\theta_l \leqslant \ \max\Big\{ & x_j\theta_l^{min} + \sum_{q=1}^{M_j} (x_j^{min} + \beta^j(q-1))\delta_{l,q}^j, \\ & x_j\theta_l^{max} + \sum_{q=1}^{M_j} (x_j^{min} + \beta^j q)(\delta_{l,q}^j - (\theta_l^{max} - \theta_l^{min})\lambda_q^j)\Big\}, \\ x_j\theta_l \geqslant \ \min\Big\{ & x_j\theta_l^{min} + \sum_{q=1}^{M_j} (x_j^{min} + \beta^j q)\delta_{l,q}^j, \\ & x_j\theta_l^{max} + \sum_{q=1}^{M_j} (x_j^{min} + \beta^j(q-1))(\delta_{l,q}^j - (\theta_l^{max} - \theta_l^{min})\lambda_q^j)\Big\}. \end{aligned} \tag{11}$$

Now, we obtain a piecewise affine approximation of the original problem by replacing bilinear terms of (6) with (2) and (11). The obtained approximation problem is an mp-MILP problem whose bilinear terms are approximate, thus it turns out to be an RHS-mp-MILP problem, which is called the tightening piecewise McCormick approximation of multi-parametric programming problem (mpPMA).

To solve RHS-mp-MILP, it is sufficient to count all possible combinations of binary variables, solve the subproblems of mp-LP, and then compare the obtained solutions [7, 12]. However, for a large number of binary variables handling this approach is an arduous task. As a result, a lot of researchers have looked for methods to decrease the number of combinations of binary variables. the branch and bound strategy [19] and the decomposition algorithm [7, 12] are two types of those methods. In this work the decomposition algorithm applies to solve RHS-mp-MILP problem. Based on the decomposition method, the problem is alternatively decomposed into mp-LP and MILP subproblems. First of all, the value of the binary variable will be fixed for a critical region. Then the resulting mp-LP should be solved (step 6), now the obtained solution provides an upper bound for the value function in that region. Secondly, for each critical region that is obtained from the first step, an MILP subproblem will be constructed (step 8) by adding the integer cuts and parametric cuts along with assuming the parameter θ as a variable (*1*). The steps of described approximation and solution based on decomposition method [12] are given in Algorithm *1*.

Algorithm 1 H [0]
Let $j \in \{1, 2, \dots, n_x\}$, $l \in \{1, 2, \dots, n_\theta\}$, N_l indicates the number of a partitions corresponding θ_l, M as a partitioning number corresponding x, $CR = \Theta$ and $z^{up}(\theta) = \infty$. Also consider x^{min} and θ^{min} lower bounds, x^{max} and θ^{max} upper bounds of x and θ respectively. Furthermore, let $BL^\theta = \{l : \exists j \in \{1, 2, \dots, n_x\} \ni (j, l) \in BL\}$. In (1) take the uncertainty parameter θ as a decision variable and solve this with a nonlinear solver and obtain the upper bound z'. (z' is obtained with solver BARON in GAMS environment.) $BL^\theta \neq \emptyset$ choos $l^* \in BL^\theta$ determine the N_{l^*} corresponding to it and define $BL_{l^*} = \{j : \ (j, l^*) \in BL\}$. $BL_{l^*} \neq \emptyset$ $j^* \in BL_{l^*}$. $n^* \leqslant N_{l^*}$ Solve minimization and maximization (5). If both problems have solutions, the tighter bounds $(x^{min}_{j^*l^*n^*}, x^{max}_{j^*,l^*n^*})$ corresponding x_{j^*} in the n^* partition is obtained. Otherwise, delete n^* corresponding θ_{l^*}.
Update the bounds of x_{j^*}: $x^{*min}_j = min_n x^{min}_{j^*l^*n^*}$, $x^{*max}_j = max_n x^{max}_{j^*l^*n^*}$ and let $BL_{l^*} = BL_{l^*} \setminus \{j^*\}$. $BL^\theta = BL^\theta \setminus \{l^*\}$
Let $x^{min}_j = x^{*min}_j$ and $x^{max}_j = x^{*max}_j$. By using (11) construct mpPMA and consider an initial integer solution $\bar{y}$. ($\bar{y}$ is an integre solution of MILP problem which is obtained from mpPMA by taking the uncertainty parameter θ as a decision variable.)
each CR with a new integer solution $\bar{y}$ Solve mp-LP problem in CR achieve both the optimal value of parametric objective function $\hat{z}_t(\theta)$ and critical region CR_t, $t = 1, 2, \dots T$, whis is obtained by substitutting $y = \bar{y}$ in (mpPMA). Establish a closed polyhedral convex region $IR_{t'}$, $t' = 1, 2, \dots T'$ with $\cup_{t'} IR_{t'} = cl(CR \backslash \cup_t CR_t)$, where the mpPMA is infeasible and within that set $\hat{z}_{t'}(\theta) = \infty$ [8]. $\hat{z}(\theta) \leqslant z^{up}(\theta)$ for some region of θ update $z^{up}(\theta)$, and corresponding integer solution [12]. myalg

Algorithm 2 [0] myalg
each region in step 5 a) formulated the MILP subproblem by considering θ as a decision variable in (mpPMA) and defining integer cuts

$$\sum_{\{k|y^{to}_k=1\}} y_k - \sum_{\{k|y^{to}_k=0\}} y_k \leqslant |J| - 1, \quad o = 1, 2, \dots O$$

and parametric cuts

$$(c + H\theta)^T x + (d + R\theta)^T y \leqslant \hat{z}_t(\theta), \quad t = 1, 2, \ldots T.$$

let $|J|$ and O respectively describes the cardinality of $J = \{k|y_k^{to} = 1\}$, and the number of integer solutions. b) solve MILP subproblem MILP subproblem is feasible the new integer solution $\bar{y}$ is found and go step 5. the final solution is given by $z^{up}(\theta)$ in the corresponding CR

The number of critical regions of the new presented method for solving (6) depends on the algorithm that is used to solve the mp-LP subproblem. In this work the and YALMIP Matlab toolbox can be used to solve the mp-LP subproblem. The solution of this mp-LP problem is obtained by dividing the parameter space into the polytope region and obtaining the optimal solution and the value function related to each of them. According to Theorem 7.10 of [8], the critical regions of the approximate RHS-mp-MILP problem is the union of a finite number of polyhedra and the value function is piecewise affine on polyhedra. If for a fixed $\theta_0 \in \Theta$ there exists a finite optimal solution, then for all $\theta \in \Theta$, the problem has either a finite optimum or no feasible solution. Also, during the mp-LP algorithm, it is necessary to compare the solutions and to calculate the rest of the space in each step. An approach for generating a polyhedra partition of the rest of the space is described in Theorem 5.2 of [8]. This procedure allows us to recursively explore the parameter space and terminate after a finite time. Therefore, the new algorithm ends up in a finite number of steps.
It should be noted that in this work, we are proposing a novel algorithm that covers uncertainties in the RHS and LHS of constraints as well as OFC uncertainties; all together and simultaneously. So far, two MATLAB toolboxes that have been presented to solve mp-MILP problems, MPT [18], POP [29], that are limited to uncertainties in RHS and OFC. Hence, they can not solve the general mp-MILP problems that is the subject of proposed algorithm. In the following, several examples are illustrated to clarify the efficiency of Algorithm *1* and the results obtained from this method are compared with the other methods. It is worth mentioning that we use Windows 7 operating system with hardware (Intel (R) Core (TM) i5, 2.5 GHz, 2GB RAM) for running the software GAMS 23.3, Matlab 2013, Cplex 12.1, and YALMIP 2015.

Remark 3.1 As the partition factor M increases, the number of critical regions in each iteration increases. Thus the partition factor has a direct impact on the computation time of the problem. One should be cautious that as the value of the partition factor increases, the depth of each iteration decreases. So, the value of M must be chosen such that a very good approximate solution yields, also the problem doesnt increase in size.

4. Examples

In this section, several examples of the mp-MILP problems with general uncertainty are presented and the computational requirements for Algorithm *1* based mpPMA model are also compared with other recent methods. Then, the application of this algorithm for larger problems is investigated using the relationship between mp-P and EMPC.

Table 2. The number of subproblems and critical regions for solving Example4.1.

	Proposed method			Algorithm from [37]		
	$MILP$	$mp-LP$	CR	$MINLP$	$mp-LP$	CR
$N=(2,2)$	12	1	3	4	1	3
$N=(3,3)$	27	2	9	17	6	11
$N=(4,4)$	35	4	19	52	17	35
$N=(8,8)$	70	11	25	150	33	117

4.1 *Numerical example*

In this subsection, the efficiency of the Algorithm *1* for solving a general mp-MILP in order to reduce the amount of calculations has been compared with another algorithm for solving these problems presented in [37] and critical regions and the optimal solution obtained from the new algorithm with critical regions and the optimal solution presented in [10] are reported.

Example 4.1 Consider the following LHS-mp-MILP problem that includes two continuous variables (x_1, x_2), two binary variables (y_1, y_2) and two uncertainty parameters (θ_1, θ_2). Besides, bilinear terms $\theta_1 x_2$ and $\theta_2 x_1$ are in left-hand side of constraints

$$
\begin{aligned}
z(\theta) &= \min_{x,y}(-2x_1 - x_2 + y_1 + y_2) \\
& x_1 + (3+\theta_1)x_2 + y_1 \leqslant 9 \\
& (2+\theta_2)x_1 + x_2 - y_2 \leqslant 8 \\
& x_1 - y_1 + y_2 \leqslant 4 \\
& 0 \leqslant x_1 \leqslant 4 \\
& 0 \leqslant x_2 \leqslant 3 \\
& y_k \in \{0,1\}, \quad k = 1,2 \\
& 0 \leqslant \theta_l \leqslant 10, l = 1,2.
\end{aligned} \tag{12}
$$

By applying the proposed method to above example, an RHS-mp-MILP approximated problem associated to this is obtained and solved. Table 2 shows the number of subproblems that must be solved, as well as the number of critical regions extracted and compared with the result of Two-Step method that is presented in [37]. The consequences made it clear that the number of critical regions and the number of subproblems that are required to solve the example 4.1 is less than the consequences of the method that is presented in [37] while partitioning number is increasing. Also, CPU time that is required for obtaining the solution of example 4.1 compared to [37] in Table 3 which shows that has decreased significantly. Furthermore, the partitioning of the parameter space and the approximate value of the objective function for $M = (2,2)$ and $M = (8,8)$ are respectively shown in Figure 1, and the optimal values of the variables, the value of the objective function and the critical region with $M = (2,2)$ by applying the new presented method and algorithm presented in [10] are reported in Table 4 and Table 5, respectively.

Example 4.1 is a typical example of the mp-MILP problem with uncertainty in the technology matrix, as the results show, the proposed approximated model of this has been successfully solved. In the following examples of mp-MILP problem with general uncertainty are given and applicability of the proposed method for these problem have been investigated.

Example 4.2 Consider the general mp-MILP problem [37]. $\theta_1 x_1$ in the objective

Table 3. Comparing CPU time for solving Example4.1 by diffrent partitioning factor.

	Proposed method (min:s)	Method reported in [37] (min:s)
$N = (2, 2)$	0 : 01	0 : 03
$N = (3, 3)$	0 : 02	0 : 07
$N = (4, 4)$	0 : 03	0 : 19
$N = (8, 8)$	0 : 03	1 : 02
$N = (16, 16)$	0 : 04	11 : 48

Table 4. Optimal solution and critical regions obtained by the new presented algorithm for Example 4.1 with partitioning number M=(2,2).

y_1	y_2	x_1	x_2	z	CR
0	0	$-0.3\theta_1 + 1.2\theta_2 + 3$	$0.6\theta_1 + 2$	$-2.4\theta_2 - 8$	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 0.8321 & -0.5547 \\ -0.2425 & 0.99701 \end{bmatrix} \leqslant \begin{bmatrix} 0 \\ 0 \\ 1.3868 \\ 0.8085 \end{bmatrix}$
0	0	$\theta_2 + 2.5$	3	$-2\theta_2 - 8$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 1 \\ -0.8321 & 0.5547 \end{bmatrix} \leqslant \begin{bmatrix} 10 \\ 0 \\ 1.5 \\ -1.3868 \end{bmatrix}$
0	0	4	3	-11	$\begin{bmatrix} 1 & -0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \leqslant \begin{bmatrix} 10 \\ 10 \\ -1.5 \\ -2.67 \end{bmatrix}$

Table 5. Optimal solution and critical regions obtained by algorithm presented in [10] for Example 4.1.

y_1	y_2	x_1	x_2	z	CR
0	0	$\frac{-8\theta_1-15}{\theta_1\theta_2+2\theta_1+3\theta_2+5}$	$\frac{-9\theta_2-10}{\theta_1\theta_2+2\theta_1+3\theta_2+5}$	$\frac{-2(-8\theta_1-15)-9\theta_2-10}{\theta_1\theta_2+2\theta_1+3\theta_2+5}$	$\begin{cases} \frac{-5}{6} \leqslant \theta_1 \leqslant 0 \\ 0 \leqslant \theta_2 \leqslant \frac{-6\theta_1-5}{3\theta_1} \\ \theta_1 \geqslant 0 \\ \theta_2 \geqslant 0 \end{cases}$
0	0	4	$-4\theta_2$	$-8 + 4\theta_2$	$\begin{cases} \theta_1 \frac{-5}{6} \\ \theta_2 \geqslant 0 \\ \frac{-5}{6} \leqslant \theta_1 \leqslant 0 \\ \theta_2 \geqslant \frac{-6\theta_1-5}{3\theta_1} \end{cases}$
0	0	$-\frac{5}{2+\theta_2}$	3	$-3 - \frac{10}{2+\theta_2}$	$\begin{cases} \theta_1 \leqslant \frac{-4}{3} \\ \frac{-3}{4} \leqslant \theta_2 \leqslant 0 \\ \theta_1 \geqslant \frac{-4}{3} \\ \frac{-5}{12+4\theta_1} \leqslant \theta_2 \leqslant 0 \end{cases}$
0	0	4	3	-11	$\begin{cases} \theta_1 \leqslant \frac{-4}{3} \\ \theta_2 \leqslant \frac{-3}{4} \end{cases}$
0	0	4	$-\frac{5}{3+\theta_1}$	$-8 - \frac{5}{3+\theta_1}$	$\begin{cases} \theta_1 \geqslant \frac{-4}{3} \\ \theta_2 \leqslant -\frac{5}{12+4\theta_1} \end{cases}$

function and $\theta_3 x_1$ on the left-hand side of the second constraint are bilinear terms,

$$\begin{aligned} z(\theta) &= \min_{x,y}((-3+\theta_1)x_1 - 8x_2 + 4y_1 + 2y_2) \\ & x_1 + x_2 + y_1 \leqslant 13 + \theta_2 \\ & (5+\theta_3)x_1 - 4x_2 \leqslant 20 \\ & -8x_1 + 22x_2 \leqslant 121 \end{aligned}$$

$$
\begin{aligned}
&-4x_1 - x_2 \leqslant -8\\
&x_1 - 10y_1 \leqslant 0\\
&x_2 - 15y_1 \leqslant 0\\
&0 \leqslant x_1 \leqslant 10\\
&0 \leqslant x_2 \leqslant 15\\
&y_k \in \{0,1\}, \quad k = 1,2\\
&0 \leqslant \theta_l \leqslant 10, l = 1,2,3.
\end{aligned}
\tag{13}
$$

Table 6 shows the number of subproblems that must be solved, as well as the number of critical regions extracted and compared with the result of Two-Step method that is presented in [37]. The consequences made it clear that the number of critical regions and the number of subproblems that are required to solve Example 4.2 is less than the consequences of the method that is presented in [37] while partitioning number is increasing. Also, CPU time that is required for this example reported in Table 7 and compared with the result of [37]. In Figure 2 you can see partitioning of the parameter space with $M = (2,2)$ and $M = (16,16)$. The approximate value of decision variables with partitioning number $M = (2,2)$ are shown in Table 8. Computational results of Example 4.2 show that the

Table 6. The number of subproblems and critical regions for Example.4.2

	Proposed method			Algorithm from [37]		
	$MILP$	$mp-LP$	CR	$MINLP$	$mp-LP$	CR
$N=(1,1)$	13	2	7	4	1	3
$N=(2,2)$	18	1	4	8	3	5
$N=(4,4)$	22	2	8	25	10	15
$N=(8,8)$	43	3	10	82	36	46
$N=(16,16)$	78	2	18	274	126	148

Table 7. Comparing CPU time for solving Example4.2 by diffrent partitioning factor.

	Proposed method (min:s)	Method reported in [37] (min:s)
$N=(1,1)$	0 : 01	0 : 05
$N=(2,2)$	0 : 01	0 : 05
$N=(4,4)$	0 : 03	0 : 21
$N=(8,8)$	0 : 03	0 : 29
$N=(16,16)$	0 : 03	2 : 06
$N=(32,32)$	0 : 03	12 : 20

Algorithm *1* is applicable for general mp-MILP problem and also reduces the size of computations. The following example show that the number of subproblems remain constant with any value of the partitioning number.

Table 8. Approximate value of decision variables of Example 4.2 with the partitioning number M=(2,2)

y_1	y_2	x_1	x_2
1	0	$-2.8205\theta_3 + 11.8426$	$-1.0256\theta_3 + 9.8077$
1	0	10	9.1364
1	0	$0.7333\theta_2 + 5.5$	$0.26667\theta_2 + 7.5$
0	1	$0.7333\theta_2 + 5.5$	$0.26667\theta_2 + 7.5$

Table 9. The number of subproblems and critical regions of the solution of Example.4.3.

	$MILP$	$mp-LP$	CR	$CPU(min:s)$
$M=(2,2)$	12	1	2	$0:01$
$M=(4,4)$	20	1	2	$0:01$
$M=(8,8)$	36	1	2	$0:01$
$M=(16,16)$	68	1	2	$0:02$
$M=(32,32)$	132	1	2	$0:04$

Example 4.3 Consider the general mp-MILP problem [10]

$$\begin{aligned} z(\theta) &= \min_{x,y}((6.4+0.25\theta_1)x_1+6x_2+(7.5+0.3\theta_1)y_1+5.5y_2) \\ & 0.8x_1+(0,67+0.015\theta_1)x_2+y_1 \geqslant 10+\theta_2 \\ & (5+\theta_3)x_1-4x_2 \leqslant 20 \\ & x_1-40y_1 \leqslant 0 \\ & x_2-40y_2 \leqslant 0 \\ & x_j \geqslant 0, j=1,2 \\ & 0 \leqslant x_2 \leqslant 40 \\ & y_j \in \{0,1\}, \quad k=1,2 \\ & -20 \leqslant \theta_l \leqslant 20, l=1,2. \end{aligned} \tag{14}$$

The optimal solution, critical regions and the number of subproblems of this example are reported in Table 9 for different partitioning numbers that last column contains the CPU time of the proposed method in the solution of 4.1. Experimental results show that by increasing partitioning number, partitioning of the parameter space does not change. The critical regions and the optimal solution of Example 4.3 by applying the proposed method as well as the algorithm presented in [10] are respectively given in Table 10 and Table 11.

4.2 *EMPC for urban traffic network*

Consider the part of urban traffic network is given in Figure 3, that model by multi-class queueing networks, where classes relate to different types of network elements [21]. Vehicles arrive from out of network, pass through Delay (D), Rout (R), and Queue (Q) Classes, then end up in Sink (S) classes.

The network has three intersections and is modeled with 22 different classes. The free flow speed is assumed at 60 km/h and each queue corresponds to a 1 km length road segment. The average vehicles length is 5 meters and the minimum distance between them in congestion conditions is 2.5 meters, so the queue capacity is 135 cars. The flow rates are measured by a long-running Simulation of Urban MObility (SUMO) [25] under heavy traffic load with arbitrary traffic signalization and the following values are reported.

$$f_1=f_2\ldots=f_{16}=45, \quad f_{17}=f_{18}=\ldots=f_{22}=20. \tag{15}$$

The average number of vehicles arriving in each cycle for $\{D_{11}^1, D_{31}^3\}$, $\{D_{31}^1, D_{11}^3\}$ and $\{D_{13}^2, D_{33}^2\}$ is 17, 13 and 7 cars/cycle that random noises have the standard deviation of 8, 6, 3 is added to this respectively. The main roads that cause congestion at the second intersection is modeled with D_{11}^1, D_{11}^2, D_{12}^2, $\ldots$, D_{12}^5 in the West-East and D_{31}^3, D_{31}^2, D_{32}^2, $\ldots$, D_{35}^2 in the NorthSouth. By this data, the following model is obtained for netwoks [21]

$$\tilde{X}_{k+1} = \tilde{X}_k + \hat{B}U_k + w_k, \tag{16}$$

$$\begin{bmatrix}\Phi^{\tilde{x}} \ \Phi^{link} \ \Phi^{light}\end{bmatrix} \begin{bmatrix}\tilde{X}_k \\ U_k^{link} \\ U_k^{light}\end{bmatrix} \leqslant \hat{b} + E^w w_k. \tag{17}$$

In above equation,vector U_k^{link} is fraction of a time unit during which each link are active, U_k^{light} green duration of traffic lights, and $\tilde{X}_k$ is a vector of number of vehicles per queue. Matrices $\hat{B}, \Phi^x, \Phi^{link}, \Phi^{light}, E^w, \hat{b}$ and w_k are defined in the appendix [21].
Without loss of generality, assume control horizon and predictive horizon are $\mathcal{N}$. First, write (17) explicitly to express all future states as a function of the future inputs $u_0, u_1, \ldots$ and then eliminate all intermediate states, consequently

$$\tilde{X}_k = \tilde{X}_0 + \sum_{q=0}^{k-1} \hat{B}U_{k-1-q} + \sum_{q=0}^{k-1} w_{k-1-q}, \quad k = 1, 2, \ldots, \mathcal{N}. \tag{18}$$

By substituting (18) to (17),

$$\Phi^u U_k + \sum_{q=0}^{k-1} \Phi^{\tilde{x}} \hat{B} U_{k-1-q} + \sum_{q=0}^{k-1} \Phi^{\tilde{x}} w_{k-1-q} \leqslant b_k - E^w w_k - \Phi^{\tilde{x}} \tilde{X}_0 \tag{19}$$

where $\Phi^u = [\Phi^{link} \quad \Phi^{light}]$, $U_k = [U_k^{link} \quad U_k^{light}]$, and $k = 1, 2, \ldots, \mathcal{N}$. Therfore MPC over the predictive horizon $\mathcal{N}$ is expressed as following optimal control problem,

$$\mathcal{J}_0^*(\tilde{X}_0) = \min_{[u_0, u_1, \ldots, u_{\mathcal{N}-1}]} \Big\{ \max_{w_0, \ldots, w_{N-1}} \Big(\sum_{k=0}^{\mathcal{N}-1} \|\bar{Q}\tilde{X}_k\|_\infty + \sum_{k=0}^{\mathcal{N}-1} \bar{R}U_k \Big) \Big\},$$
$$subj.\ to$$
$$\begin{cases} \tilde{X}_k = \tilde{X}_0 + \sum_{q=0}^{k-1} \hat{B}U_{k-1-q} + \sum_{q=0}^{k-1} w_{k-1-q} \\ \Phi^u U_k + \sum_{q=0}^{k-1} \Phi^{\tilde{x}} \hat{B} U_{k-1-q} + \sum_{q=0}^{k-1} \Phi^{\tilde{x}} w_{k-1-q} \leqslant b + E^w w_k - \Phi^{\tilde{x}} \tilde{X}_0 \\ k = 1, 2, \ldots, \mathcal{N} - 1, \quad \tilde{X}_0 = \tilde{X}(t_0) \end{cases}$$

where $\bar{Q}$ is a matrix where every element is equal to one and is assumed to be a matrix with negative weights of traffic flow on each link [21]. The obtainded result reported in [21] by the following date show that MPC controller with this cost function may reduce the congestion inside the network. By solving (20), the input sequence $U^* = [U_0^*, U_1^*, \ldots, U_{\mathcal{N}-1}^*]$ is obtained, then by applying the first element of U^* to (18), satat at the next time step is calculated. The optimization problem (20) is repeated at next time step, and this process is repeated recursively along predictive horizon based on the new state. Therefore, by apply MPC, it is necessary to solve the $\mathcal{N}$ optimal control problem online, which shows the computational complexity of this method. Using mp-P can reduce the complexity of MPC online computing to offline.To achieve this, the auxiliary variables $\epsilon_0^{\tilde{x}}, \ldots, \epsilon_{\mathcal{N}-1}^{\tilde{x}}$ are used to linearize infinite norm in cost function of (20) according to [8], then

$$\mathcal{J}_0^*(\tilde{X}(0)) = \min_{[u_0,u_1,\ldots,u_{\mathcal{N}-1},\epsilon_1^{\tilde{x}},\ldots,\epsilon_N^{\tilde{x}}]} \Big\{ \max_{w_0,\ldots,w_{\mathcal{N}-1}} \Big(\sum_{k=1}^{\mathcal{N}-1} \epsilon_k^{\tilde{x}} + \sum_{k=1}^{\mathcal{N}-1} \bar{R}U_k \Big) \Big\}, \tag{20}$$

$$subj.\ to$$

$$\begin{cases} \epsilon_k^{\tilde{x}} \leqslant \pm\bar{Q}\tilde{X}_k \\ \tilde{X}_k = \tilde{X}_0 + \sum_{q=0}^{k-1} \hat{B}U_{k-1-q} + \sum_{q=0}^{k-1} w_{k-1-q} \\ \Phi^u U_k + \sum_{q=0}^{k-1} \Phi^{\tilde{x}} \hat{B}U_{k-1-q} + \sum_{q=0}^{k-1} \Phi^{\tilde{x}} \hat{B} w_{k-1-q} \leqslant b + E^w w_k - \Phi^{\tilde{x}} \tilde{X}_0 \\ k = 1, 2, \ldots, \mathcal{N}-1 \\ \tilde{X}_0 = \tilde{X}(t_0) \end{cases}$$

As stated before, for MPC along $\mathcal{N}$, should be solve $\mathcal{N}$ optimization problem in form (20) for each initial state $\tilde{X}_0$, but if we consider $\tilde{X}_0$ as a parameter in eq. (20), we have a mp-LP problem and the explicit optimal solution of this will be an affine function of $\tilde{X}_0$ [8]. In other words, solution of (20) is obtained as a function of $\tilde{X}_0$.

In (17), the traffic flow rates are variable parameter and be adjusted on traffic measurements. These parameter are assumed known and constant (15) in [21], therefore $\hat{B}$ are fixed and kown and the complexity of (20) are reduced. Now, if we consider the traffic flow rates as parameters, according to the definition of matrix $\hat{B}$, RHS- mp-LP problem in (20) convert to a RHS-LHS-mp-LP problem and the resulting problem can be solved by Algorithm *1*. In this way, while significantly reducing the the computational requirment, a more general case of urban traffic control is also examined. In the following, MPC of [21] and new EMPC are applied to urban traffic network is given in Figure 3 and the results are compared.
MPC controller is designed in [21] and EMPC by using (20) and Algorithm *1* is also applied to this example with predictive horizon $\mathcal{N} = 10$ and the results are reported in Figure 4. Two traffic routes are effective in congestion at the second intersection, the main traffic flows of queues $D_{11}^1 - Q_{11}^1$, $D_{31}^3 - Q_{31}^3$ and the side traffic flows of queues $D_{11}^3 - Q_{11}^3$ and queues $D_{31}^1 - Q_{31}^1$. The simulation results show that both MPC and EMPC try to reduce the queue length in the main and side traffic (Figure 4.c and Figure 4.d) and thus prevent traffic congestion at the second intersection (Figure 4.b). Figure 4.a shows that EMPC compare to MPC has no effect on improving network throughpu but Table 12 shows that its online computing time is significantly reduced.

5. Conclusions

In this paper, we propose a novel method for the solutions of an mp-MILP problem with a general uncertainty by combining tighten McCormick relaxation with multi-parametric programming. It seems that this combination has not been used for general mp-MILP problem. The algorithm consists of two steps. In the first step, we approximate bilinear terms of the coefficient matrix by taking advantage of the McCormick relaxation and the bounds of partitioning variables have been tightened with the help of linear programming. Therefore, this makes the feasible region smaller. Secondly, we apply the new bounds that are obtained from the first step. Then by presenting a piecewise linear approximation the problem changes to a multi-parametric programming problem, whose uncertainty only appears to the right-hand side of constraints, and the approximation problem is solved by using an efficient algorithm.

In fact, by optimizing the bounds of variables that appear in the form of bilinear terms, we obtained an approximate RHS-mp-MILP, this help to reduce the number of sub-problems that is mandatory to be solved for solving the main problem, and consequently the computational requirement decreases.
It is important to be cautious that as the partitioning number increases the accuracy of the approximation increases, but its value cannot be chosen as large as possible. Given that the higher value of the partitioning number, the higher number of binary variables of the approximate model, increasing the partitioning number would result in a more complex problem. Hence, one should set the value of the partitioning number in a trade-off between accuracy and complexity. In examples, we set the value in an attempt to preserve the reliability of the approximate presented model results subject to our computational limits. Also, example of urban traffic netwok with two uncertainty parameter is selected and design EMPC for it by using the proposed alghorithm. Computational results of example show that the proposed method significantly reduced online cpu time. It is important that due to the uncertainty on the flow rates, the proposed method is able to coordinate the green time split and turning fractions at intersections aiming to minimize the number of vehicles in the controlled area and also prevents form traffic jams inside network. According to the results, it seems that this method can be used in real traffic networks that we intend to examine this in future research.

Appendix A. Appendix

Deffinition of matrices in (16) and (17) that is given in [21].

$$T_d = \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0
\end{bmatrix}
\qquad
T_s = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0
\end{bmatrix}$$

$$\Phi = \begin{bmatrix} \overbrace{zeros(22,12)}^{\Phi^{x}} & \overbrace{-eye(22)}^{\Phi^{u,link}} & \overbrace{zeros(22,6)}^{\Phi^{u,light}} \\ zeros(612) & zeros(6,22) & -eye(6) \\ zeros(22,12) & eye(22) & zeros(22,6) \\ zeros(3,12) & zeros(3,22) & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ zeros(3,12) & zeros(3,22) & \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \\ zeros(3,12) & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} zeros(3,4) & \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \\ -eye(12) & T_s * \hat{F} & zeros(12,6) \\ eye(12) & (Td - Ts) * \hat{F} & zeros(12,6) \end{bmatrix}$$

$$B = \left[(T_d - T_s) * \hat{F} \; zeros(12,6) \right]$$

$$E^{w} = \left[zeros(size(\Phi,1) - 12, 12) \; -eye(12,12) \right]$$

$$\hat{b} = \left[zeros(size(\Phi,1) - 12, 1) \; 135 * ones(12,1) \right]$$

References

[1] Aboudolas, K., Papageorgiou, M., and Kosmatopoulos, E. (2009). Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174.

[2] Aboudolas, K., Papageorgiou, M., Kouvelas, A., and Kosmatopoulos, E. (2010). A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 18(5):680 – 694.

[3] Avis, D. and Fukuda, K. (1992). A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(3):295–313.

[4] Avraamidou, S. and Pistikopoulos, E. N. (2017). A multiparametric mixed-integer bi-level optimization strategy for supply chain planning under demand uncertainty. *IFAC-PapersOnLine*, 50(1):10178 – 10183.

[5] Avraamidou, S. and Pistikopoulos, E. N. (2019a). Multi-parametric global optimization approach for tri-level mixed-integer linear optimization problems. *Journal of Global Optimization*, 74(3):443–465.

[6] Avraamidou, S. and Pistikopoulos, E. N. (2019b). A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. *Computers & Chemical Engineering*, 125:98 – 113.

[7] Borrelli, F. (2003). *Constrained Optimal Control of Linear and Hybrid Systems*. Springer-Verlag Berlin Heidelberg.

[8] Borrelli, F., Bemporad, A., and M.Morari (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press.

[9] Castro, P. (2015). Tightening piecewise mccormick relaxations for bilinear problems. *Computers & Chemical Engineering*, 72:300–311.

[10] Charitopoulos, V. M., Papageorgiou, L. G., and Dua, V. (2018). Multi-parametric mixed integer linear programming under global uncertainty. *Computers & Chemical Engineering*, 116:279 – 295.

[11] Dinkelbach, W. (1969). *Sensitivittsanalysen und parametrische Programmierung*. Springer-Verlag Berlin Heidelberg.

[12] Dua, V. and Pistikopoulos, E. N. (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99:123–139.

[13] Faísca, N. P., Kouramas, K. I., Saraiva, P. M., Rustem, B., and Pistikopoulos, E. N. (2008). A multi-parametric programming approach for constrained dynamic programming problems. *Optimization Letters*, 2:267–280.

[14] Flavell, R. and Salkin, G. R. (1975). An approach to sensitivity analysis. *Journal of the Operational Research Society*, 26(4):857–866.

[15] Gounaris, C. E., Misener, R., and Floudas, C. A. (2009). Computational comparison of piecewiselinear relaxations for pooling problems. *Industrial & Engineering Chemistry Research*, 48(12):5742–5766.

[16] Hao, Z., Boel, R., and Li, Z. (2018). Model based urban traffic control, part ii: Coordinated model predictive controllers. *Transportation Research Part C: Emerging Technologies*, 97:23 – 44.

[17] Henderson, H. and Searle, S. (1981). On deriving the inverse of a sum of matrices. *SIAM Review*, 23(1):53–60.

[18] Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-parametric toolbox 3.0. *2013 European Control Conference,*, pages 502–510.

[19] Jia, Z. and Ierapetritou, M. (2006). Uncertainty analysis on the righthand side for milp problems. *AIChE Journal*, 52:2486 – 2495.

[20] Khalilpour, R. and Karimi, I. (2014). Parametric optimization with uncertainty on the left hand side of linear programs. *Computers & Chemical Engineering*, 60:31 – 40.

[21] Le, T., Vu, H. L., Nazarathy, Y., Vo, Q. B., and Hoogendoorn, S. (2013). Linear-quadratic model predictive control for urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 36:498 – 512.

[22] Li, Z. and Ierapetritou, G. M. (2007). A new methodology for the general multiparametric mixed-integer linear programming (milp) problems. *Industrial & Engineering Chemistry Research*, 46(15):5141–5151.

[23] Li, Z. and Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 32(4):715 – 727.

[24] Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan.

[25] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE.

[26] Lu, K., Du, P., Cao, J., Zou, Q., He, T., and Huang, W. (2019). A novel traffic signal split approach based on explicit model predictive control. *Mathematics and Computers in Simulation*, 155:105 – 114.

[27] McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems. *Mathematical Programming*, 10(1):147–175.

[28] Mitsos, A. and Barton, P. I. (2009). Parametric mixed-integer 01 linear programming: The general case for a single parameter. *European Journal of Operational Research*, 194(3):663–686.

[29] Oberdieck, R., Diangelakis, N., Papathanasiou, M., Nascu, I., and Pistikopoulos, E. (2016). Pop - parametric optimization toolbox. *Industrial & Engineering Chemistry Research*, 55(33):8979–8991.

[30] Oberdieck, R., Diangelakis, N. A., and Pistikopoulos, E. N. (2017). Explicit model predictive control: A connected-graph approach. *Automatica*, 76:103 – 112.

[31] Oberdieck, R. and Pistikopoulos, E. N. (2015). Explicit hybrid model-predictive control: The exact solution. *Automatica*, 58:152 – 159.

[32] Pistikopoulos, E. N., Dua, V., Bozinis, N. A., Bemporad, A., and Morari, M. (2000). On-line optimization via off-line parametric optimization tools. *Computers & Chemical Engineering*, 24(2):183 – 188.

[33] Ryu, J., Dua, V., and Pistikopoulos, E. N. (2007). Proactive scheduling under uncertainty: a parametric optimization approach. *Industrial & Engineering Chemistry Research*, 46(24):8044–8049.

[34] Sakizlis, V., Kakalis, N. M., Dua, V., Perkins, J. D., and Pistikopoulos, E. N. (2004). Design of robust model-based controllers via parametric programming. *Automatica*, 40(2):189 – 201.

[35] Winkler, B. (1998). *Grbner Bases and Applications*. London Mathematical Society Lecture Note Series. Cambridge University Press.

[36] Wittmann-Hohlbein, M. and Pistikopoulos, E. N. (2012). A two-stage method for the approximate solution of general multiparametric mixed-integer linear programming problems. *Industrial & Engineering Chemistry Research*, 51(23):8095–8107.

[37] Wittmann-Hohlbein, M. and Pistikopoulos, E. N. (2014). Approximate solution of mp-milp problems using piecewise affine relaxation of bilinear terms. *Computers & Chemical Engineering*, 61:136 – 155.

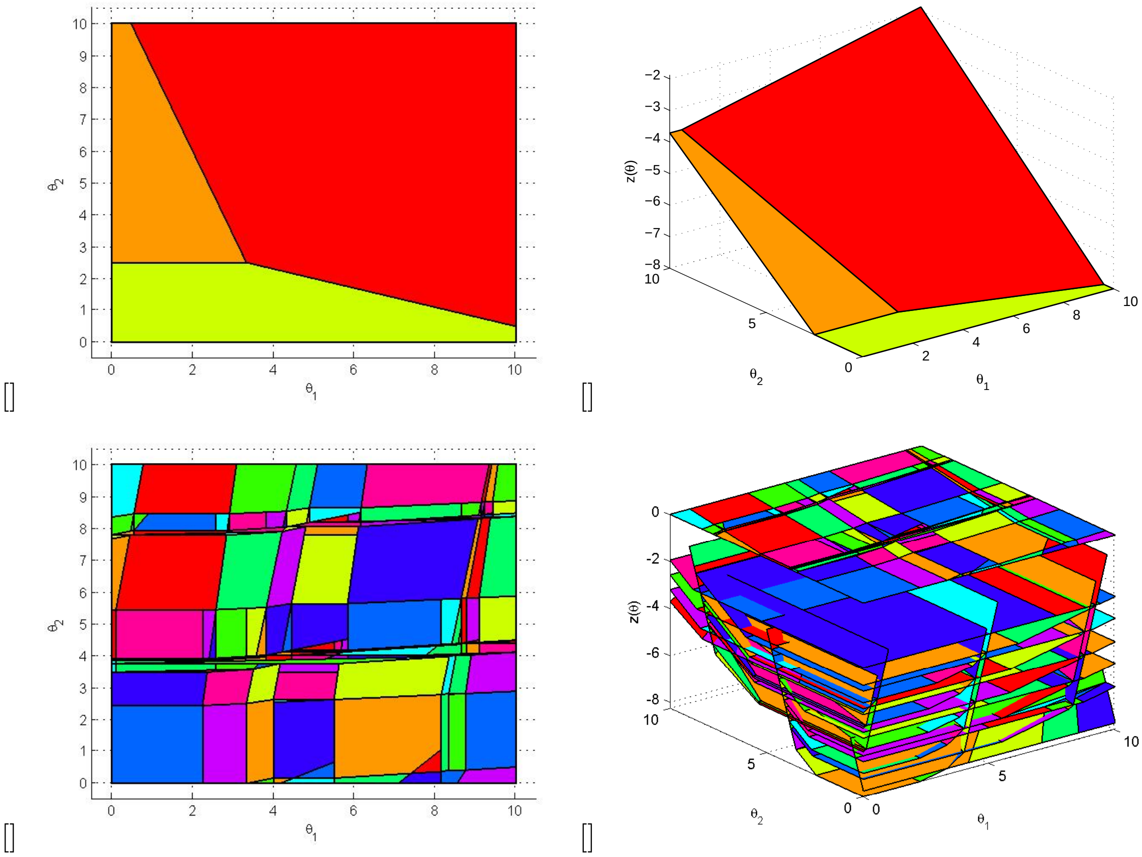


Figure 1. a) Partitioning of parameter space with $M = (2, 2)$, b) Approximate objective function with $M = (2, 2)$, c) Partitioning of parameter space with $M = (4, 4)$, d) Approximate objective function with $M = (4, 4)$ for Example 4.1.

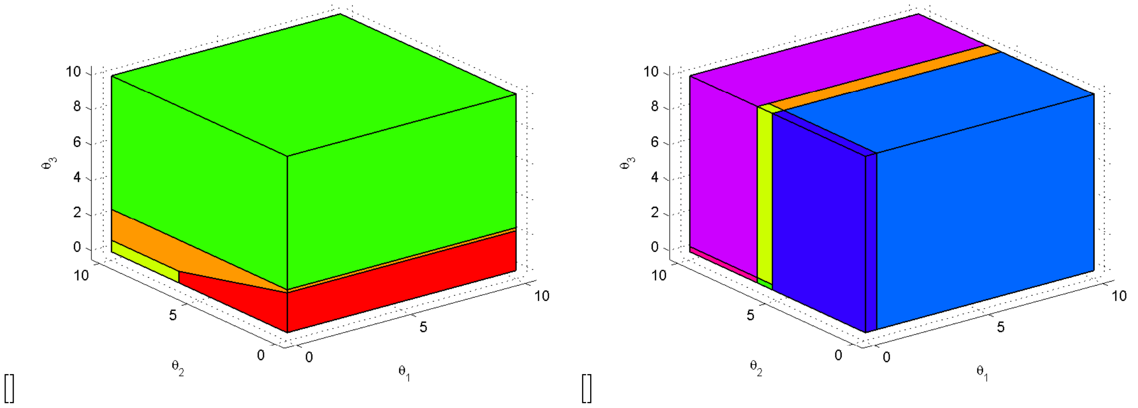


Figure 2. a) Partitioning of the parameter space with $M = (2, 2)$, b) Partitioning of the parameter space with $M = (16, 16)$

Table 10. Optimal solution and critical regions obtained by proposed method for Example. 4.3 witth any partitioning number.

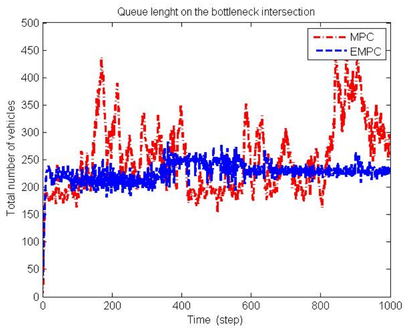
y_1	y_2	x_1	x_2	CR
1	0	0	0	$\begin{cases} -20 \leqslant \theta_1 \leqslant 20 \\ -20 \leqslant \theta_2 \leqslant -10 \end{cases}$
1	0	$1.25\theta_2 + 12.5$	0	$\begin{cases} -20 \leqslant \theta_1 \leqslant 20 \\ -10 \leqslant \theta_2 \leqslant 20 \end{cases}$

Table 11. Optimal solution and critical regions obtained by algorithm presented in [10] for Example.4.3.

y_1	y_2	x_1	x_2	CR
1	0	0	0	$\begin{cases} -20 \leqslant \theta_1 \leqslant 20 \\ -20 \leqslant \theta_2 \leqslant -10 \end{cases}$
0	1	0	$\frac{66.67\theta_2+666.67}{\theta_1+44.67}$	$\begin{cases} 0 \leqslant \theta_1 \leqslant 20 \\ -10 \leqslant \theta_2 \leqslant -8 \\ -0.0675 \leqslant \theta_1 \leqslant 0 \\ -8 \leqslant \theta_2 \leqslant \frac{\theta_1(-10.96\theta_1-751.947)+1079.47}{\theta_1(\theta_1+70.267)-136.533} \end{cases}$
1	0	$1.25\theta_2 + 12.5$	0	$\begin{cases} \begin{cases} -20 \leqslant \theta_1 \leqslant 0.0675 \\ -10 \leqslant \theta_2 \leqslant 20 \end{cases} \\ \begin{cases} -0.0675 \leqslant \theta_1 \leqslant 0 \\ -10 \leqslant \theta_2 \leqslant -8 \\ \frac{\theta_1(-10.96\theta_1-751.947)+1079.47}{\theta_1(\theta_1+70.267)-136.533} \leqslant \theta_2 \leqslant 20 \end{cases} \\ \begin{cases} 0 \leqslant \theta_1 \leqslant -20 \\ -8 \leqslant \theta_2 \leqslant 20 \end{cases} \end{cases}$

Figure 3. Traffic network with queue classes and with input flow rate in D^1_{11} D^2_{13} D^3_{31} D^2_{33} D^1_{31} and D^3_{11}

[Total vehicles out from the network] [Queue length at secound



intersection]

[Queue length of main traffic] [Queue length of side traffic]

Figure 4. Compare the impact of MPC and EMPC on queue length and network throughput

time step	EMPC	MPC
	min:s	min:s
200	0 : 5	23 : 36
500	0 : 13	8 : 15
1000	0 : 31	7 : 34

Table 12. Online cpu time computting of MPC and EMPC per (minutes, second)