Solving a Signalized Traffic Intersection Problem with NLP Solvers

Teófilo Miguel M. Melo*, João Luís H. Matias†, M. Teresa T. Monteiro‡

*CIICESI, School of Technology and Management of Felgueiras, Polytechnic of Porto, Portugal,
E-mail:tmm@estgf.ipp.pt

† Centre of Mathematics, CM-UTAD, University of Trás-os-Montes e Alto Douro, Portugal,
E-mail:j_matias@utad.pt

‡ R&D Centre Algoritmi, Department of Production and Systems, University of Minho, Portugal,
E-mail:tm@dps.uminho.pt

Abstract

Mathematical Programs with Complementarity Constraints (MPCC) finds many applications in areas such engineering design, economic equilibrium and mathematical theory itself. In this work we consider a queuing system model resulting from a single signalized traffic intersection regulated by pre-timed control in an urban traffic network. The model is formulated as an MPCC problem and may be used to ascertain the optimal cycle and the green split allocation. This MPCC problem is also formulated as its NLP equivalent reformulation. The goal of this work is to solve the problem, using both MPCC and NLP formulations, minimizing two objective functions: the average queue length over all queues and the average waiting time over the worst queue. The problem was codified in AMPL and solved using some optimization software packages.

Keywords: traffic control, complementarity constraints, NLP, SQP

1 Introduction

Mathematical Programs with Complementarity Constraints (MPCC) is a subclass of more general Mathematical Programs with Equilibrium Constraints (MPEC). These kind of constraints may come as a game, a variational inequality or as stationary conditions of an optimization problem. The main applications areas are Engineering and Economics [1, 2, 3]. They are so widespread in these areas because the concept of complementarity is synonymous with the notion of system equilibrium. They are very difficult to solve as the usual constraint qualifications necessary to guarantee the algorithms convergence fail in all feasible points [4]. This complexity is caused by the disjunctive nature of the complementarity constraints.

Some nonlinear approaches to solve MPCC have been proposed, starting with the smoothing scheme [5, 6], the regularization scheme [7, 8], the interior point methods [9], the penalty approaches [10, 11, 12] and the "elastic mode" for nonlinear programming in conjunction with a sequential quadratic programming (SQP) algorithm [13]. In Fletcher *et al.* [14] the quadratic convergence of SQP is guaranteed, near a stationary point, under relatively mild conditions.

As the number of vehicles and the need for transportation grow, traffic light control can be used to control the flow of the traffic in urban environments. Schutter and Moor [15] study the optimal traffic control problem of a two two-way streets intersection. These authors derive an approximate model that describes the evolution of the queues lengths as a continuous function of time.

Starting from this model it is possible to compute the traffic light switching scheme that minimizes a criterion such as average queue length over all queues, the average waiting time over the worst queue or average waiting time.

A single signalized intersection regulated by pre-timed control problem with four traffic streams is considered. This problem is formulated as an MPCC and also as its NLP equivalent reformulation. Some computational experiments using the FilterMPEC, the KNITRO and the MATLAB optimization toolbox are performed.

This paper is organized as follows. Next section defines the MPCC problem and its NLP reformulation. Some optimal issues are presented in Section 3. The traffic model formulation using two objective functions is described in Section 4. In Section 5, the optimization solvers main characteristics are presented. Numerical experiments using the solvers are reported in Section 6. Some conclusions and future work are carried out in Section 7.

2 Problem Definition

We consider Mathematical Program with Complementarity Constraints (MPCC):

$$\begin{array}{ll} \min & f(x) \\ s.t. & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \\ & 0 \leq x_1 \perp x_2 \geq 0, \end{array} \tag{MPCC}$$

where f and c are the nonlinear objective function and the constraint functions, respectively, assumed to be twice continuously differentiable. E and I are two disjoined finite index sets with cardinality p and m, respectively. A decomposition $x = (x_0, x_1, x_2)$ of the variables is used where $x_0 \in \mathbb{R}^n$ (control variables) and $(x_1, x_2) \in \mathbb{R}^{2q}$ (state variables). The expressions $0 \leq x_1 \perp x_2 \geq 0 : \mathbb{R}^{2q} \to \mathbb{R}^q$ are the q complementarity constraints. Q is a finite index set with cardinality q.
One attractive way of solving (MPCC) is to replace the complementarity constraints by a set of nonlinear inequalities, such as $x_{1j}\, x_{2j} \leq 0,\ j \in Q$, and then solve the equivalent nonlinear program (NLP):

$$\begin{array}{ll} min & f(x) \\ s.t. & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \\ & x_{1j}x_{2j} \leq 0,\ j \in Q, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array} \tag{MPCC-NLP}$$

3 Optimal Issues

The NLP formulation has no feasible point that strictly satisfies the inequalities. This fact implies that the Mangasarian-Fromovitz constraint qualification (MFCQ) is violated at every feasible point [16]. This failure has consequences: the multiplier set is unbounded, the central path fail to exist, the active constraints normals are linearly dependent and linearizations of the NLP formulation can be inconsistent arbitrarily close to the solution. Recent developments show that there is a relationship between strong stationarity defined by Scheel and Scholtes and the Karush-Kuhn-Tucker (KKT) points. This relationship established convergence of SQP methods for MPCC formulated as NLP. Some optimality concepts used in this work are based on the study of Fletcher *et al.*[14].

4 Traffic Model Formulation

On this section we briefly describe the traffic model and present the MPCC formulation - further details could be consulted in the work of Ribeiro and Simões [17]. The traffic model considers an intersection with four traffic streams, S_1, S_2, S_3 and S_4 which are controlled by a traffic signal, T_1, T_2, T_3 and T_4 respectively. A set $S = \{1, 2, 3, 4\}$ is considered. The intersection presented in Figure 1 is controlled by

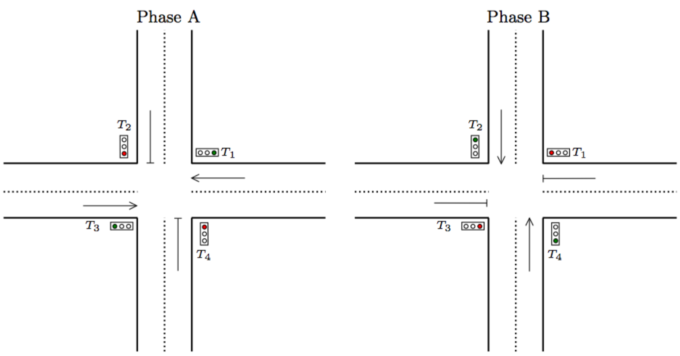

Figure 1: Intersection.

two phases (A and B). During the Phase A, the traffic signals T_1 and T_3 have green light and the same

occurs in Phase B for T_2 and T_4. In both phases, the cycle has 3 stages: green, yellow and red. The arrival rate of vehicles in traffic stream S_i at instant time t is $\lambda_i(t)$ for $i \in S$. When the traffic signal T_i is green, the departure rate in traffic stream S_i at instant time t is $\mu_i(t)$ and in the case of the traffic being yellow, the departure rate in traffic stream S_i at instant t is $\kappa_i(t)$ for $i \in S$. Let $t_0, t_1, \ldots$, be the time instants when a change in the traffic signal occurs.
The duration of the yellow time and clearance time is fixed and set equal to d_Y and d_C respectively. The time instants when traffic signals T_1 and T_3 initiate a green period and T_2 and T_4 begin a red period are $t_0, t_2, t_4, \ldots$. The time instants when traffic signals T_1 and T_3 initiate a red period and T_2 and T_4 begin a green period are $t_1, t_3, t_5, \ldots$. Thus, $t_{2k+1} - t_{2k} = y_G + d_Y + d_C$ and $t_{2k+2} - t_{2k+1} = y_R + d_Y + d_C$, $k \in \mathbb{N}_0$.
Therefore, y_G represents the green time in traffic signals T_1 and T_3 and y_R represents the red time at T_1 and T_3 and a cycle length is equal to $y_G + y_R + 2d_y + 2d_C$.
The traffic problem has the following MPCC formulation:

$$\begin{array}{ll} \min & \displaystyle\sum_{i=1}^{4} \left(\frac{1}{2N}(x_0)_i + \sum_{k=1}^{N-1} \frac{1}{N}(x_k)_i + \frac{1}{2N}(x_N)_i \right) \\ s.t. & 0 \leq x_k \leq x_{max}, \\ & y_{min} \leq y_R \leq y_{max}, \\ & y_{min} \leq y_G \leq y_{max}, \\ & x_{2k+1} \geq x_{2k} + b_1 y_G + b_3 \perp x_{2k+1} \geq b_5, \\ & x_{2k+2} \geq x_{2k+1} + b_2 y_R + b_4 \perp x_{2k+2} \geq b_6, \end{array} \tag{TP1}$$

where, the objective function represents the average queue length over all queues. Since short cycles imply more stops and long cycles causes long delays, maximum and minimum durations for the red and green time (y_R and y_G) are considered as a simple bound constraints. The number of vehicles in traffic stream i at time instant k is represented by $(x_k)_i$. The maximum queue length in each traffic stream is x_{max} and N is the time periods considered.

Another situation was considered and has the following formulation:

$$\begin{array}{ll} \min & \displaystyle\max_i \left(\frac{1}{\bar{\lambda}_i} \left(\frac{1}{2N}(x_0)_i + \sum_{k=1}^{N-1} \frac{1}{N}(x_k)_i + \frac{1}{2N}(x_N)_i \right) \right) \\ s.t. & 0 \leq x_k \leq x_{max}, \\ & y_{min} \leq y_R \leq y_{max}, \\ & y_{min} \leq y_G \leq y_{max}, \\ & x_{2k+1} \geq x_{2k} + b_1 y_G + b_3 \perp x_{2k+1} \geq b_5, \\ & x_{2k+2} \geq x_{2k+1} + b_2 y_R + b_4 \perp x_{2k+2} \geq b_6. \end{array} \tag{TP2}$$

In this case, the goal is to minimize the average waiting time over the worst queue and the constraints are the same as the traffic problem (TP1).

In addition, the following vectors are defined:

$$\begin{array}{l} x_k = [L_1(t_k),\ L_2(t_k),\ L_3(t_k),\ L_4(t_k)]^T,\ k \in \mathbb{N}_0, \\ b_1 = [\bar{\lambda}_1 - \bar{\mu}_1,\ \bar{\lambda}_2,\ \bar{\lambda}_3 - \bar{\mu}_3,\ \bar{\lambda}_4]^T, \\ b_2 = [\bar{\lambda}_1,\ \bar{\lambda}_2 - \bar{\mu}_2,\ \bar{\lambda}_3,\ \bar{\lambda}_4 - \bar{\mu}_4]^T, \\ b_3 = [(\bar{\lambda}_1 - \bar{\kappa}_1)d_Y + \bar{\lambda}_1 d_C,\ \bar{\lambda}_2(d_C + d_Y),\ (\bar{\lambda}_3 - \bar{\kappa}_3)d_Y + \bar{\lambda}_3 d_C,\ \bar{\lambda}_4(d_C + d_Y)]^T, \\ b_4 = [\bar{\lambda}_1(d_C + d_Y),\ (\bar{\lambda}_2 - \bar{\kappa}_2)d_Y + \bar{\lambda}_2 d_C,\ \bar{\lambda}_3(d_C + d_Y), (\bar{\lambda}_4 - \bar{\kappa}_4)d_Y + \bar{\lambda}_4 d_C]^T, \\ b_5 = [\max\{(\bar{\lambda}_1 - \bar{\kappa}_1)d_Y + \bar{\lambda}_1 d_C,\ \bar{\lambda}_1 d_C\},\ 0,\ \max\{(\bar{\lambda}_3 - \bar{\kappa}_3)d_Y + \bar{\lambda}_3 d_C,\ \bar{\lambda}_3 d_C\},\ 0]^T, \\ b_6 = [0,\ \max\{(\bar{\lambda}_2 - \bar{\kappa}_2)d_Y + \bar{\lambda}_2 d_C,\ \bar{\lambda}_2 d_C\},\ 0,\ \max\{(\bar{\lambda}_4 - \bar{\kappa}_4)d_Y + \bar{\lambda}_4 d_C,\ \bar{\lambda}_4 d_C\}]^T, \end{array}$$

where for each traffic stream, *i.e*, for $i \in S$:

- $\bar{\lambda}_i$ is the average arrival rate,
- $\bar{\mu}_i$ is the average departure rate when the traffic signal is green,
- $\bar{\kappa}_i$ is the average departure rate when the traffic signal is yellow,
- $L_i(t_k)$ is the queue length at time instant k.

These two MPCC problems, (TP1) and (TP2), are reformulated into their NLP equivalent formulation (MPCC-NLP). Next section presents three optimization codes used to solve these two traffic situations in both formulations (MPCC) and (MPCC-NLP).

5 NLP Solvers

In this section we present three solvers, two of them available on the NEOS Server platform [18, 19], the third one is available on MATLAB software. The NEOS Server is a free internet-based service for solving optimization problems with several solvers representing the state of the art in optimization software.

5.1 FilterMPEC

FilterMPEC arises from the work of Fletcher and Leyffer [20] to solve mathematical programs with equilibrium constraints (MPECs) in AMPL format [21]. FilterMPEC is an extension of filterSQP, implementing a SQP solver which is suitable for solving medium scale nonlinearly constrained problems.

5.2 KNITRO

KNITRO was developed for the solution of general nonconvex, nonlinearly constrained optimization problems in AMPL format [22]. It is also effective for problems with complementarity constraints. KNITRO provides three algorithms, two of them use the interior point methods, the other is based on an active set method.

5.3 MATLAB

The `fmincon` routine from MATLAB optimization toolbox [23] is a gradient-based method to solve problems with objective and constraint functions twice continuously differentiable. It uses one of four algorithms: active-set, interior point method, SQP or trust-region-reflective. From these algorithms, we use the SQP algorithm to solve (MPCC-NLP). To connect the modelling language AMPL [24] to the `fmicon` routine, a MATLAB mex function was used.

6 Numerical Experiments

This section summarizes the results of the numerical tests, solving the traffic intersection problem with two different approaches: the first approach, used by filterMPEC and KNITRO, the traffic problem is solved using the (MPCC) formulation, the second approach employed by `fmincon` routine uses (MPCC-NLP) reformulation. The MATLAB version was 7.11.0 (R2010b) and the computational experiments were made on a 2.26 GHz Intel Core 2 Duo with 8GB of RAM, MAC OS 10.6.8 operating system. For all the solvers, the default options were used.
Both traffic situations (TP1) and (TP2) were tested using ten instances of $\bar{\lambda}_i,\ i \in S$, corresponding to problems P1-P10 reported in Table 1. Furthermore, $N = 61$ was considered, which corresponds to a period of time over thirty cycles. The other parameters used are:

$$\begin{array}{lll} \bar{\mu}_i = 1800\, veh/h & \bar{\kappa}_i = 1800\, veh/h & i \in S \\ x_{0k} = 2\%\bar{\lambda}_k & x_{0j} = 1\%\bar{\lambda}_k & \text{for } k = 1,\ 3,\ \ j = 2,\ 4 \\ x_{max_i} = 25 & & i \in S \\ d_Y = 3\ s & d_C = 2\ s & \\ y_{min} = 7 \text{ s} & y_{max} = 60 \text{ s} & \end{array}$$

Table 1: 10 instances

Problem	$\bar{\lambda}_1$ (veh/h)	$\bar{\lambda}_2$ (veh/h)	$\bar{\lambda}_3$ (veh/h)	$\bar{\lambda}_4$ (veh/h)
P1	150	850	250	750
P2	150	800	250	700
P3	500	900	600	800
P4	500	850	600	750
P5	550	900	650	800
P6	650	800	750	700
P7	300	750	400	650
P8	450	600	550	500
P9	700	750	800	650
P10	850	600	850	500

Table 2 and Table 3 report, the optimization results of traffic problem (TP1) and traffic problem (TP2), respectively: the red split time (y_R), the green split time (y_G) and objective function value (obj) for the three solvers.

Table 2: (TP1) results

	FilterMPEC			KNITRO			fmincon		
Problem	y_R (s)	y_G (s)	obj	y_R (s)	y_G (s)	obj	y_R (s)	y_G (s)	obj
P1	26,5	7,0	7,1	11,6	7,0	11,7	26,5	7,0	6,9
P2	23,6	7,0	6,6	13,7	7,0	7,1	23.6	7,0	6.4
P3	31,1	18,7	17,5	20,7	13,8	24,8	31,1	18,7	17,2
P4	28,8	17,7	15,2	17,2	12,2	17,9	28,8	17,7	14,9
P5	37,3	24,8	21,2	34,5	23,2	21,1	37,3	24,8	20,9
P6	32,0	27,9	21,0	20,4	18,0	23,0	32,0	27,9	20,7
P7	16,8	7,1	7,8	13,16	7,0	7,6	16,8	7,1	7,6
P8	13,5	10,6	8,32	10,9	9,8	8,0	13,5	10,6	8,0
P9	29,8	29,7	20,8	25,9	26,8	22,1	29,8	29,7	20,4
P10	21,3	27,9	16,6	21,3	27,9	16,2	21,3	27,9	16,2

Table 3: (TP2) results

	FilterMPEC			KNITRO			fmincon		
Problem	y_R (s)	y_G (s)	obj	y_R (s)	y_G (s)	obj	y_R (s)	y_G (s)	obj
P1	18,8	7,0	14,5	13,3	7,0	17,2	18,8	7,0	14,5
P2	17,4	7,0	13,8	17,4	7,0	13,8	17,4	7,0	13,8
P3	28,2	16,4	26,5	19,9	12,4	39,7	39,0	22,7	24,5
P4	30,4	18,4	21,6	15,2	10,1	35,1	32,9	19,6	21,6
P5	36,4	24,0	40,2	20,8	14,0	75,9	41,4	27,0	29,9
P6	31,2	26,7	33,3	25,6	22,5	45,2	39,0	32,6	31,6
P7	15,0	7,2	14,3	14,1	7,0	15,1	16,2	7,9	14,2
P8	11,6	9,5	15,4	11,9	9,6	15,2	15,3	11,6	14,6
P9	31,0	30,0	32,9	20,1	20,3	65,8	35,9	34,0	32,3
P10	22,0	26,7	25,4	20,6	27,4	29,1	24,2	28,5	25,4

Table 4: Number of iterations

	FilterMPEC		KNITRO		fmincon	
	it		it		it	
Problem	(TP1)	(TP2)	(TP1)	(TP2)	(TP1)	(TP2)
P1	5	68	58	159	21	30
P2	5	68	18	204	24	25
P3	2	32	26	21	10	47
P4	8	54	41	106	43	58
P5	2	81	53	171	20	77
P6	3	84	34	118	15	61
P7	2	34	19	48	37	38
P8	4	49	15	70	25	59
P9	2	53	29	174	6	41
P10	3	66	45	81	24	99

Table 4 presents the number of iterations (it) achieved by the solvers for both traffic situations (TP1) and (TP2). Among the analysed solvers, filterMPEC is the one that presents the best performance solving the traffic situation (TP1). For traffic situation (TP2), KNITRO needs more iterations than the others solvers.

7 Conclusions and Future Work

A traffic model was studied and codified in AMPL language which could be easily connected to nonlinear programming solvers. Two approaches were used to solve the traffic problem: the (MPCC) formulation, solved by FilterMPEC and KNITRO on NEOS Server and the reformulation (MPCC-NLP) solved by `fmicon` routine from MATLAB. The model for traffic situations (TP1) and (TP2) was efficiently solved by these solvers using both approaches (MPCC) and (MPCC-NLP).

As a first stage of future work we intend to introduce a probabilistic distribution in arrival rate instead of consider it as a constant value. Another idea is to allow that vehicles can turn (right or left) and not follow just ahead. With minor changes other objective functions can be easily tested: average queue length over the worst queue, worst case queue length and average waiting time over all queues. Another goal of this work is to collect real data from a signalized traffic intersection in a city in order to test the model in a real situation.

Acknowledgement

This work is funded by FEDER funds through Operational Programme for Competitiveness Factors - COMPETE and National Funds through FCT - Foundation for Science and Technology in Project scope: FCOMP-01-0124-FEDER-022674 (R&D unit Algoritmi). The work is also supported by PEst-OE/MAT/UI4080/2011. The authors are very grateful to all the financial support.

References

[1] Luo, Z., Pang, J. and Ralph, D. (1996). Mathematical programs with equilibrium constraints. Cambridge University Press.

[2] Ferris, M. and Pang, J. (1997). Engineering and economic applications of complementarity constraints. *SIAM Review*, 39(4):669–713.

[3] Outrata, J., Kocvara, M. and Zowe, J. (1998). Nonsmooth approach to optimization problems with equilibrium constraints. Kluwer Academic Publishers, Dordrecht.

[4] Chen, X. and Florian, M. (1995). The nonlinear bilevel programming problem: formulations, regularity and optimality conditions. *Optimization*, 32(4):193–209.

[5] Facchinei, F., Jiang, H. and Qi, L. (1996). A smoothing method for mathematical programs with equilibrium constraints. Technical Report 03.96, Universita di Roma, "La Sapienza".

[6] Fukushima, M. and Pang, J. (1999). Convergence of a smoothing continuation method for mathematical programs with complementarity constraints. Lectures Notes in Economics and Mathematical Systems, Springer-Verlag, 447:99–110.

[7] Scholtes, S. (2001). Convergence proprieties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 11(4):918–936.

[8] Monteiro, M., and Rodrigues, H. (2011). Combining the regularization strategy and the SQP to solve MPCC – A MATLAB implementation. *Journal of Computational and Applied Mathematics*, 235(18):5348–5356.

[9] Raghunathan, A. and Biegler, L. (2005). An interior point method for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(3):720-750.

[10] Hu, X. and Ralph, D. (2004). Convergence of a penalty method for mathematical programs with complementarity constraints. *Journal of Optimization Theory and Applications*, 123(2):365–390.

[11] Ralph, D. and Wright, S. (2004). Some proprieties of regularization and penalization schemes for MPECs. *Optimization Methods and Software*, 19(5):527–556.

[12] Monteiro, M., and Meira, J. (2011). A penalty method and a regularization strategy to solve MPCC. *International Journal of Computer Mathematics*, 88(1):145–149.

[13] Anitescu, M. (2005). On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236.

[14] Fletcher, R., Leyffer, S., Ralph, D. and Scholtes, S. (2006). Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 17(1):259–286.

[15] Schutter, B. and Moor, B. (1998). Optimal traffic light control for a single intersection. *European Journal of Control*, 4(3)260–276.

[16] Scheel, H. and Scholtes, S. (2000). Mathematical programs with complementarity constraints: Stationarity, optimality and sensitivity. *Mathematics of Operations Research*, 25(1):1–22.

[17] Ribeiro, I. and Simões, M. (2012). Optimal cycle for a signalized intersection using global optimization and complementarity. *TOP*, 20(3):777–790.

[18] Neos Server (2013). URL: `http://www.neos-server.org/neos/`

[19] Dolan, E., Fourer, R., Moré, J., and Munson, T. (2002). The NEOS Server for Optimization Version 4 and Beyond. Preprint ANL/MCS-TM-253, Mathematics and Computer Science Division, Argonne National Laboratory.

[20] Fletcher, R. and Leyffer, S. (2003). Solving Mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 19(1):15–40.

[21] Fourer, R. and Kernighan, B. (1993). AMPL: A modeling language for mathematical programming. Duxburg Press, Massachusetts.

[22] Byrd, R., Nocedal, J., and Waltz, R. (2006). KNITRO: An integrated package for nonlinear optimization. In: Large-scale nonlinear optimization. Springer US, 2006. p. 35-59.

[23] Moler, C., Little, J., and Bangert, S. (2001). The MathWorks, Sherborn, Mass. Matlab User's Guide - The Language of Technical Computing.

[24] Gay, M. (1997). Hooking Your Solver to AMPL. Technical report, Bell Laboratories, Murray Hill, NJ.