Solving a signalized traffic intersection problem with an hyperbolic penalty function

Teófilo Melo, M. Teresa T. Monteiro, and João Matias

Citation: AIP Conf. Proc. **1479**, 830 (2012); doi: 10.1063/1.4756266
View online: http://dx.doi.org/10.1063/1.4756266
View Table of Contents: http://proceedings.aip.org/dbt/dbt.jsp?KEY=APCPCS&Volume=1479&Issue=1
Published by the AIP Publishing LLC.

Solving a Signalized Traffic Intersection Problem with an Hyperbolic Penalty Function

Teófilo Melo*, M. Teresa T. Monteiro† and João Matias**

*CIICESI, School of Technology and Management of Felgueiras, Polytechnic of Porto, Portugal
†Algoritmi Centre, Department of Production and Systems, University of Minho, Portugal
**Centre of Mathematics, CM-UTAD, Portugal

Abstract. Mathematical Program with Complementarity Constraints (MPCC) finds many applications in fields such as engineering design, economic equilibrium and mathematical programming theory itself. A queueing system model resulting from a single signalized intersection regulated by pre-timed control in traffic network is considered. The model is formulated as an MPCC problem. A MATLAB implementation based on an hyperbolic penalty function is used to solve this practical problem, computing the total average waiting time of the vehicles in all queues and the green split allocation. The problem was codified in AMPL.

Keywords: Penalty technique, Traffic control, MPCC, NLP
PACS: 87.55.de

INTRODUCTION

Mathematical Program with Complementarity Constraints is an exciting new application of nonlinear techniques. In Engineering, problems dealing with contact, obstacle and friction, process modeling, deformation and traffic congestion are treated. An important reason why complementarity optimization problems are so important in Engineering is because the concept of complementarity is tantamount with the notion of system equilibrium. They are very difficult to solve as the usual constraints qualifications, necessary to guarantee the algorithms convergence, fail in all feasible points. This complexity is caused by the disjunctive constraints. Researchers have been studied the MPCC theory and proposing efficient algorithms - one can emphasize the work of Fukushima and Pang [1], Scholtes [2], Anitescu [3], Ralph [4] and Fletcher *et al.* [5]. The interior point method (IPM), the sequential quadratic programming (SQP), the smooth nonlinear programming, the penalty technique and regularization scheme are some strategies that have been studied to implement numerical algorithms. Recent studies of Scheel and Scholtes [6] have proved that the strong stationarity of an MPCC equals the first order optimality conditions of the NLP equivalent. This fact motivates the scientific community to use NLP approaches to deal with MPCC. Some relevant works are [7], [8], [9] and [10].

As the number of vehicles and the need for transportation grow, traffic light control can be used to augment the flow of the traffic in urban environments. Schutter and Moor [11], study the optimal traffic control problem for an intersection of two two-way streets. They 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, worst case queue length or average waiting time. The main difference on this approach and the most existing methods is that the model allows the green-amber-red cycle time to vary from one cycle to another.

A signalized intersection regulated by pre-timed control problem with four traffic lanes is considered. This problem is formulated as an MPCC problem and solved by the algorithm combining the SQP and the hyperbolic penalty technique [12].

This paper is organized as follows. We start presenting the MPCC formulation and the hyperbolic penalty technique. Next section defines the traffic problem to solve. The results are reported in the Computational Experiments section. Finally some conclusions are carried out.

Numerical Analysis and Applied Mathematics ICNAAM 2012
AIP Conf. Proc. 1479, 830-833 (2012); doi: 10.1063/1.4756266

MPCC 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. One attractive way of solving (MPCC) is to replace the complementarity constraints by a set of nonlinear inequalities, such as $x_{1j}x_{2j} \leq 0, j = 1, \ldots, q$, and then solve the equivalent nonlinear program (NLP):

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

HYPERBOLIC PENALTY TECHNIQUE

A way to deal with the complementarity constraints is to apply a penalty technique. In this work an hyperbolic penalty function presented by Xavier [13] is used to penalize the complementarity constraints ($x_{1j}x_{2j} \leq 0,\ j = 1, \ldots, q$) in (NLP):

$$P(x, u, v) = f(x) + P_t \qquad \text{and} \qquad P_t = \sum_{j=1}^{q} -u(x_{1j}x_{2j}) + \sqrt{u^2(x_{1j}x_{2j})^2 + v^2}$$

where P_t is the penalty term and u, v are parameters with $u, v \geq 0$, $u \to \infty, v \to 0$. It is a two phase penalty approach: in the first stage, the initial parameter u increases, thus causing a reduction in the penalty to the points outside the feasible region while at the same time there is a reduction in the penalty for the points inside the feasible region. This phase continues until a feasible point is obtained. From this point on, u remains constant and the values of v decrease sequentially. In this approach, the complementarity terms are penalized and a sequence of the following nonlinear constrained optimization problem is solved as far u is incremented and v decreased:

$$\begin{array}{ll} min & P(x, u, v) \\ s.t. & c_i(x) = 0, i \in E, \\ & c_i(x) \geq 0, i \in I, \\ & x_1 \geq 0,\ \ x_2 \geq 0, \end{array} \tag{PEN}$$

The feasibility test carried out to x_k, to update the penalty parameters is as below

$$\begin{cases} v_{k+1} = v_k\rho_2,\ 0 < \rho_2 < 1, & \textbf{if } x_{1j}x_{2j} > \dfrac{-v_k}{1000}\ ,\ j = 1, \ldots, q. \\ u_{k+1} = u_k\rho_1,\ \rho_1 > 1, & \textbf{otherwise} \end{cases}$$

TRAFFIC MODEL FORMULATION

The traffic problem to be solved has the following MPCC formulation:

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

where, the objective function represents the total average waiting time experienced by vehicles in all queues. The problem formulation assumes that the duration of the yellow time (d_Y) and clearance time (d_C) is fixed. 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) have been also added to the problem constraints. The number of vehicles in lane i at time instant k is represented by $(x_k)_i$. The maximum queue length in each traffic stream is x_{max}, N is the time periods considered. In addition, the following vectors are defined [14]:

$$
\begin{array}{l}
x_k = [L_1(t_k),\, L_2(t_k),\, L_3(t_k)],\, L_4(t_k)]^T,\, k \in \mathbb{N}_0 \\
b_1 = [\bar{\lambda}_1 - \bar{\mu}_1,\, \bar{\lambda}_2,\, \bar{\lambda}_3 - \bar{\mu}_3,\, \bar{\lambda}_4]^T \\
b_2 = [\bar{\lambda}_1,\, \bar{\lambda}_2 - \bar{\mu}_2,\, \bar{\lambda}_3,\, \bar{\lambda}_4 - \bar{\mu}_4]^T \\
b_3 = [(\bar{\lambda}_1 - \bar{\kappa}_1)d_Y + \bar{\lambda}_1 d_C,\, \bar{\lambda}_2(d_C + d_Y),\, (\bar{\lambda}_3 - \bar{\kappa}_3)d_Y + \bar{\lambda}_3 d_C,\, \bar{\lambda}_4(d_C + d_Y)]^T \\
b_4 = [\bar{\lambda}_1(d_C + d_Y),\, (\bar{\lambda}_2 - \bar{\kappa}_2)d_Y + \bar{\lambda}_2 d_C,\, \bar{\lambda}_3(d_C + d_Y), (\bar{\lambda}_4 - \bar{\kappa}_4)d_Y + \bar{\lambda}_4 d_C]^T \\
b_5 = [\max\{(\bar{\lambda}_1 - \bar{\kappa}_1)d_Y + \bar{\lambda}_1 d_C,\, \bar{\lambda}_1 d_C\},\, 0,\, \max\{(\bar{\lambda}_3 - \bar{\kappa}_3)d_Y + \bar{\lambda}_3 d_C,\, \bar{\lambda}_3 d_C\},\, 0]^T \\
b_6 = [0,\, \max\{(\bar{\lambda}_2 - \bar{\kappa}_2)d_Y + \bar{\lambda}_2 d_C,\, \bar{\lambda}_2 d_C\},\, 0,\, \max\{(\bar{\lambda}_4 - \bar{\kappa}_4)d_Y + \bar{\lambda}_4 d_C,\, \bar{\lambda}_4 d_C\}]^T
\end{array} \tag{2}
$$

where for each lane, *i.e*, for $i = 1, \ldots, 4$:

- $\bar{\lambda}_i$ 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.

COMPUTATIONAL EXPERIMENTS

This section summarizes the results of the computational tests solving (1) with the hyperbolic penalty algorithm. The problem was tested with nine sets of parameters, corresponding to nine problems (P1-P9). Table 1 reports parameters for each test problem, the four arrival rates and the results of the optimization process, namely the green split time (y_G), the red split time (y_R) and the cycle length. The cycle length, in the last column, is defined by: $y_G + y_R + 2(d_C + d_Y)$ (see [14] for more details).

TABLE 1. Results

Problem	$\bar{\lambda}_1$ (veh/h)	$\bar{\lambda}_2$ (veh/h)	$\bar{\lambda}_3$ (veh/h)	$\bar{\lambda}_4$ (veh/h)	y_G (s)	y_R (s)	Cycle (s)
P1	150	850	250	750	7	19.7	36.7
P2	150	800	250	700	7	17.5	34.5
P3	500	900	600	800	18.3	30.3	58.6
P4	500	850	600	750	17.1	27.3	54.4
P5	550	900	650	800	23.9	35.7	69.6
P6	650	800	750	700	27.9	32	69.9
P7	300	750	400	650	7.2	15.7	32.9
P8	450	600	550	500	10.3	12.5	32.8
P9	700	750	800	650	29.7	28.8	69.5

The parameters used are:

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

CONCLUSIONS AND FUTURE WORK

A signalized traffic intersection model formulated as an MPCC problem was solved using a MATLAB iterative algorithm based on the hyperbolic penalty function. The numerical results show that the hyperbolic penalty function performs well solving the problem. In addition, the cycle length and green split results are very close to the ones achieved by Ribeiro *et al.* [14]. The problem was codified in AMPL and is free available in MacMPEC database [15] named TrafficSignalCycle. As a future work, we intend to encode four new versions, changing the objective function - average queue length over all queues, average queue length over the worst queue, average waiting time over the worst queue and the worst case queue length.

ACKNOWLEDGEMENT

Research co-funded by the Portuguese Government through the FCT - Fundação para a a Ciência e a Tecnologia under the project PEst-OE/MAT/UI4080/2011.

REFERENCES

1. M. Fukushima, and J. Pang, *Lecture Notes in Economics and Mathematical Systems* **447**, 99–110 (1999).
2. S. Scholtes, *SIAM Journal on Optimization* **11**, 918–936 (2001), ISSN 10526234, URL `http://link.aip.org/link/SJOPE8/v11/i4/p918/s1&Agg=doi`.
3. M. Anitescu, *SIAM Journal on Optimization* **15**, 1203–1236 (2005), ISSN 10526234.
4. D. Ralph, *Philosophical Transactions of the Royal Society A-Mathematical Physical and Engineering Sciences* **366**, 1973–987 (2008).
5. R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, *SIAM Journal on Optimization* **17**, 259 (2006), ISSN 10526234.
6. H. Scheel, and S. Scholtes, *Mathematics of Operations Research* **25**, 1–22 (2000).
7. M. Anitescu, P. Tseng, and S. J. Wright, *Mathematical Programming* **110**, 337–371 (2006), ISSN 0025-5610, URL `http://www.springerlink.com/index/10.1007/s10107-006-0005-4`.
8. D. Ralph, and S. Wright, *Optimization Methods and Software* **19**, 527–556 (2004), ISSN 1055-6788.
9. R. Fletcher, and S. Leyffer, *Optimization Methods and Software* **19**, 15–40 (2004), ISSN 1055-6788, URL `http://www.informaworld.com/openurl?genre=article&doi=10.1080/10556780410001654241&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3`.
10. M. T. T. Monteiro, and J. F. Meira, *International Journal of Computer Mathematics* **88**, 145–149 (2011), ISSN 0020-7160.
11. B. D. Schutter, *Proceedings of the American Control Conference* pp. 2195–2199 (1999).
12. T. Melo, M. T. T. Monteiro, and J. Matias, *AIP Conference Proceedings (ICNAAM)* **1389**, 763–766 (2011).
13. A. E. Xavier, *International Transactions in Operational Research* **8**, 659–671 (2001), ISSN 0969-6016.
14. I. M. Ribeiro, and M. L. Simões, *Sociedad de Estadística e Investigación Operativa* (2010).
15. S. Leyffer, MacMPEC (2012), URL `http://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC`.