QVILIB: A LIBRARY OF QUASI-VARIATIONAL INEQUALITY TEST PROBLEMS

Francisco Facchinei[1], Christian Kanzow[2], and Simone Sagratella[1]

Preprint 308 March 2012

[1] Sapienza University of Rome
Department of Computer, Control, and
Management Engineering
Via Ariosto 25, 00185 Roma
Italy
e-mail: facchinei@dis.uniroma1.it
sagratella@dis.uniroma1.it

[2] University of Würzburg
Institute of Mathematics
Emil-Fischer-Str. 30, 97074 Würzburg
Germany
e-mail: kanzow@mathematik.uni-wuerzburg.de

March 28, 2012

Abstract. Quasi-variational inequalities (QVIs) are a well established and important modelling tool with several applications in different fields. Solution methods exist, but are either especially tailored to very particular classes of problems, or are investigated mainly from a theoretical point of view. However, spurred by several recent applications of QVIs in engineering, the interest in general purpose solution methods is growing. The aim of this paper is to present a collection of test problems from diverse sources which gives a uniform basis on which algorithms for the solution of QVIs can be tested and compared. All the proposed problems are implemented in Matlab and the collection is freely available on request.

Key Words: Quasi-variational inequalities; Test problem; Walrasian equilibrium problem; Generalized Nash equilibrium problem; Contact problem.

1 Introduction

Given a function $F : \mathbb{R}^n \to \mathbb{R}^n$ and a point-to-set mapping $K : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, the (finite-dimensonal) quasi-variational inequality problem (QVI for short) is to find a point $x^* \in K(x^*)$ such that

$$F(x^*)^T(y - x^*) \geq 0, \qquad \forall y \in K(x^*). \tag{1}$$

In the particular case where the set K is independent of x and therefore constant, the QVI reduces to the well-known variational inequality problem (VI for short), see, e.g. [5] for a comprehensive treatment. However, the fact that $K(x)$ typically depends on x makes the QVI significantly more difficult, both from a theoretical and a numerical point of view.

Despite the fact that QVIs arise quite frequently and naturally in many fields, the numerical solution of (finite-dimensional or discretized infinite-dimensional) problems is still "in its infancy at best", see [14]. This is certainly due to the great difficulty of the problem. And yet, in recent years there has been a renewed interest in the optimization community that has lead to interesting advancements, see for example [4, 6, 7, 10, 11, 12, 14, 15]. However, with the exception of [4], the numerical results reported in these papers are very limited and in most cases the numerical experiences are carried out on problems that are different from paper to paper and difficult to duplicate. To overcome this state of affairs, and building on the test problem collection used in the preparation of [4], in this paper we present a collection of 55 QVI test problems. The problems range from small (few variables and constraints) to large (several thousands of variables and constraints). They include academic problems, problems arising from real-world applications (e.g. Walrasian equilibrium problems) and problems resulting from the discretization of infinite-dimensional QVIs modelling diverse engineering and physical problems. For each problem we provide a succinct, but complete description, along with Matlab files which allow the user to easily obtain function values and derivatives and that can be easily incorporated in any solution routine developed in order to solve QVIs. It is hoped that the availability of this collection, which we plan to maintain and enlarge, will stimulate the development of new solution methods and will permit a uniform and fair comparison of existing and future algorithms. The collection can freely be obtained by writing to `facchinei@dis.uniroma1.it`, `kanzow@mathematik.uni-wuerzburg.de` or `sagratella@dis.uniroma1.it`.

The paper is organized as follows. In the next section we give a general overview of how the library is organized and a detailed description of the Matlab routines and of how they can be used. In particular, we describe the format in which the QVIs are represented and show how to extract the necessary data (function values, gradients, Jacobians, or Hessians) from the implementation of each test problem. The test problems themselves together with some relevant information are given in Section 3. Note that a subset of these test problems were already used to obtain the numerical results in the authors' paper [4].

2 Test Problem Library

This section describes several classifications of the test problems to be given in the next section. It also discusses the details of the possible inputs and outputs that are available for the test problem files.

2.1 Classification of Test Problems

Each QVI, see (1), is defined by the function F and the point-to-set mapping $K(x)$. We assume that $K(x)$ is defined as the intersection of a fixed set $\bar{K}$ and a set $\tilde{K}(x)$ that depends on the point x:

$$K(x) = \bar{K} \cap \tilde{K}(x).$$

The sets $\bar{K}$ and $\tilde{K}(x)$ are described by inequalities and equalities:

$$\bar{K} := \{y \in \mathbb{R}^n \mid g^I(y) \leq 0,\, M^I y + v^I = 0\},$$

$$\tilde{K}(x) := \{y \in \mathbb{R}^n \mid g^P(y,x) \leq 0,\, M^P(x)y + v^P(x) = 0\}.$$

The constraints defining the set $\bar{K}$ are individual constraints that are independent of x, hence we use the superscript "I" in our notation (for individual/independent of x). On the other hand, the constraints defining $\tilde{K}(x)$ are parametric due to their dependence on x, therefore, we use the superscript "P" (for parametric). We assume that $g^I(\cdot)$ is a vector of convex functions and that each component function of $g_i^P(\cdot, x)$ is convex for all x. When we refer to the whole set of inequality or (linear) equality constraints, we use the notation

$$g(y,x) := \begin{pmatrix} g^I(y) \\ g^P(y,x) \end{pmatrix}, \qquad M(x)y + v(x) := \begin{pmatrix} M^I \\ M^P(x) \end{pmatrix} y + \begin{pmatrix} v^I \\ v^P(x) \end{pmatrix}.$$

For each test problem, we therefore report F and the functions defining $K(x)$ along with few more information concerning origin of the problem, known characteristics (for example monotonicity of F, uniqueness of the solution, etc). Furthermore, in some cases we also give some more details on the construction of the test problem (for example in the case of a discretization of an infinite-dimensional problem).

Each problem in the test set comes with a simple problem classification which we explain below. A problem is classified by the label [XXX-X-n-m_I-p_I-m_P-p_P]. The first character in the label defines the type of the operator F. Possible values are:

L : F is linear (L = linear)

N : F is nonlinear (N = nonlinear).

The second character in the label defines the type of constraints g^I of the problem. Possible values are:

A : there are no constraints g^I (A = absent)

B : g^I defines only bounds on the variables (B = box/bound)

L : g^I is linear (L = linear)

Q : g^I is quadratic (Q = quadratic)

N : g^I is nonlinear (N = nonlinear).

The third character in the label defines the type of constraints g^P of the problem. The classification below is based on the classes of constraints analyzed in [4], which are briefly recalled below. We refer the interested reader to [4] for a more complete discussion of these classes of constraints. Here we only note that they include most type of constraints considered in the literature and have proven to be meaningful when it comes to the analysis of algorithms. Possible values are:

A : there are no constraints g^P (A = absent)

B : g^P defines separable box (in the y-part) constraints only: a generic constraint has the form $ay_i + b(x) - c \leq 0$ (B = box/bound)

L : g^P defines separable linear (in the y-part) constraints only: a generic constraint has the form $a^T y + b(x) - c \leq 0$ (L = linear)

O : g^P defines constraints different from any of the above (O = other).

The character immediately following the first hyphen indicates the primary origin and/or interest of the problem. Possible values are:

A : the problem is academic, that is, has been constructed specifically by researchers to test one or more algorithms (A = academic)

R : the problem models some real problem: economic, physical, etc. (R = real)

D : the problem is the discretization of an infinite-dimensional problem (D = discretized).

The numbers after the second hyphen indicate the "dimensions" of the problem, in particular:

- n is the number of variables;

- m_I is the number of inequality constraints defining $\bar{K}$;

- p_I is the number linear equalities in $\bar{K}$;

- m_P is the number of inequality constraints defining $\tilde{K}(x)$;

- p_P is the number of equalities in the definition of $\tilde{K}(x)$.

In Table 1 we report the list of all problems currently in the library, with the corresponding labels.

Academic problems	
Problem name	**Label**
OutZ40 - OutZ41	[LBB-A-2-4-0-2-0]
OutZ42	[LBB-A-4-4-0-4-0]
OutZ43 - OutZ44	[LAB-A-4-0-0-4-0]
MovSet1A - MovSet1B - MovSet2A - MovSet2B	[LAO-A-5-0-0-1-0]
MovSet3A1 - MovSet3B1	[LAO-A-1000-0-0-1-0]
MovSet3A2 - MovSet3B2	[LAO-A-2000-0-0-1-0]
MovSet4A1 - MovSet4B1	[LAO-A-400-0-0-801-0]
MovSet4A2 - MovSet4B2	[LAO-A-800-0-0-1601-0]
Box1A - Box1B	[LAB-A-5-0-0-10-0]
Box2A - Box2B - Box3A - Box3B	[NAB-A-500-0-0-1000-0]
BiLin1A - BiLin1B	[LBO-A-5-10-0-3-0]
RHS1A1 - RHS1B1 - RHS2A1 - RHS2B1	[LAL-A-200-0-0-199-0]
Problems from real-world models	
Problem name	**Label**
WalEq1	[LBO-R-18-18-1-5-0]
WalEq2	[NBO-R-105-105-1-20-0]
WalEq3	[LBO-R-186-186-1-30-0]
WalEq4	[NBO-R-310-310-1-30-0]
WalEq5	[NBO-R-492-492-1-40-0]
Wal2	[NLO-A-105-107-0-20-0]
Wal3	[LLO-R-186-188-0-30-0]
Wal5	[NLO-A-492-494-0-40-0]
LunSS1	[NBA-R-501-1002-0-0-6]
LunSS2	[NBA-R-1251-2502-0-0-6]
LunSS3	[NBA-R-5001-10002-0-0-6]
LunSSVI1	[NBA-R-501-1002-1-0-0]
LunSSVI2	[NBA-R-1251-2502-1-0-0]
LunSSVI3	[NBA-R-5001-10002-1-0-0]
Discretized problems	
Problem name	**Label**
Scrim11	[LBA-D-2400-2400-0-0-1200]
Scrim12	[LBA-D-4800-4800-0-0-2400]
Scrim21	[LBL-D-2400-2400-0-2400-0]
Scrim22	[LBL-D-4800-4800-0-4800-0]
OutKZ31	[LBB-D-62-62-0-62-0]
OutKZ41	[LBB-D-82-82-0-82-0]
KunR11 - KunR21 - KunR31	[LAO-D-2500-0-0-2500-0]
KunR12 - KunR22 - KunR32	[LAO-D-4900-0-0-4900-0]

Table 1: Problem list.

2.2 Description of Matlab Functions

Each QVI test problem described in the next section is distributed as a single Matlab M-file function, whose name is the same as that of the problem. For some of the larger problems, a data file is also necessary which, again, has the same name as the problem (see below). All these files are contained in a folder called QVILIB. For each problem, and given two points y and x, the quantities given in Table 2 can be computed.

`<QVI_name>(0)`	initializes the data that are used when invoking `<QVI_name>` with other flags; does not return anything. In particular sets, as global variables, the following "dimensions": `nVar`: number of variables `nIneq`: total number of inequality constraints `nEq`: total number of equality constraints `nIneqInd`: number of inequality constraints independent of x `nEqInd`: number of equality constraints independent of x
`<QVI_name>(1,x)`	returns $F(x)$
`<QVI_name>(2,x)`	returns $JF(x)$, the Jacobian of F at x
`<QVI_name>(3,x,y)`	returns $g(y,x)$
`<QVI_name>(4,x,y)`	returns $J_y g(y,x)$, the partial Jacobian of g with respect to y
`<QVI_name>(5,x)`	returns $Jh(x)$, the Jacobian of $h(x) := g(x,x)$
`<QVI_name>(6,x)`	returns $Js_i(x)$ for all i, the Jacobian of all functions $s_i(x) := J_y g_i(y,x)\|_{y=x}$
`<QVI_name>(7,x,y)`	returns $M(x)y + v(x)$
`<QVI_name>(8,x)`	returns $M(x)$
`<QVI_name>(9,x)`	returns $Jt(x)$, the Jacobian of $t(x) := M(x)x + v(x)$
`<QVI_name>(10,x)`	returns $J(M_{i*}(x)^T)$ for all i, where $M_{i*}(x)$ denotes the ith row of the matrix $M(x)$
`<QVI_name>(11)`	clears all problem data from memory; does not return anything

Table 2: Description of all possible calls to a generic QVI function in the library.

Let us give some more explanations. To this end, consider a generic problem whose name is `<QVI_name>`; in the folder QVILIB one can find the M-file `<QVI_name>.m` and, for some of the problems, a second data file `<QVI_name>.dat`. The function `<QVI_name>` can have up to three input arguments. The first input argument of `<QVI_name>` is a mandatory flag and it is used by the user to select the behavior of the function as detailed in the previous list.

In this list, it is also shown how many additional input arguments should be used in correspondence to each admissible value for the flag `i`, which can take any integer value between 0 and 11. If the flag value `i` is out of range or if the number of input arguments is not correct an exception will be thrown. Note that, if present, the second and third input argument of `<QVI_name>` must be column vectors with `nVar` elements; otherwise, a corresponding exception will be thrown.

When the function is called with the first argument equal to 0, some preliminary operations are performed, in particular in the scope of the function some global variables are initialized. This set of global variables always contains the positive integer `nVar`, i.e. the number of variables, the nonnegative integer `nIneq`, i.e. the total number of inequality constraints, the nonnegative integer `nEq`, i.e. the total number of equality constraints, the nonnegative integer nIneqInd, i.e. the number of inequality constraints independent of x, and the nonnegative integer nEqInd, i.e. the number of equality constraints that do not depend on x. In order to make these quantities available to the user's calling function, one should define them as global also in the user's calling function(s). When the function is called with the first argument equal to 0, other global variables might be defined that store data used by the function when invoked with other flags. All these further variables begin with the string "QVItest" and therefore it is better to avoid the use of any quantities which includes this string in the user's functions. <QVI_name>(0) must be called before any other function call. If this rule is not respected, an exception will be thrown. If it is called more than one time, a warning will be displayed since unnecessary operations are performed.

When the function is called with the first argument equal to 11, then all global variables in the scope of the function will be cleared. If used, it must be the last function call.

The function <QVI_name>.m can have one output, or no output at all, depending on the value of the flag i. When present, the output can be either a column vector, a sparse matrix or a cell array of sparse matrices. Table 3 summarizes in detail all possible output formats.

Input Flag	**Output**
i= 0 or i= 11	no output
i= 1	column vector of dimension nVar
i= 2	sparse square matrix of dimensions nVar × nVar
i= 3	column vector of dimension nIneq if nIneq= 0 the output is the empty matrix
i= 4 or i= 5	sparse matrix of dimension nIneq × nVar if nIneq= 0 the output is the empty matrix
i= 6	cell array of dimension nIneq, each cell in the array contains a sparse square matrix of dimension nVar × nVar (the matrices contained in the cells are the evaluations of $Js_i(x)$) if nIneq= 0 the output is the empty cell array
i= 7	column vector of dimension nEq if nEq= 0 the output is the empty matrix
i= 8 or i= 9	sparse matrix of dimension nEq × nVar if nEq= 0 the output is the empty matrix
i= 10	cell array of dimension nEq, each cell in the array contains a sparse square matrix of dimension nVar × nVar (the matrices contained in the cells are the evaluations of $J(M_{i*}(x)^T)$) if nEq= 0 the output is the empty cell array

Table 3: Description of outputs of a generic QVI function in the library.

We already observed that, in order to help the user debugging, some simple checks are performed when the <QVI_name>.m function is invoked. If these checks fail, a corresponding error message is provided by throwing an exception or a warning. Some of these have been mentioned already; we report the complete list in Table 4.

Event	**Exception/warning thrown**
`i` is not an integer between 0 and 11	`QVItest:BadFlagInput`
`QVI_name` is invoked with `i` between 1 and 10 before invoking it with `i` = 0	`QVItest:DataNotInitialized`
`QVI_name`(0) is invoked more than once	`QVItest:MultipleDataInitialization` (warning)
`QVI_name` is invoked with a wrong number of arguments	`QVItest:BadInputNumber`
the second or third argument of `QVI_name` have wrong dimensions	`QVItest:BadInputArgument`

Table 4: Description of exceptions.

The library also includes an M-file `startingPoints.m` that can be used by the user to get the starting points of each test problem. If the function `startingPoints.m` is called without any input arguments, it displays a list of all test problems available with a brief description of their starting points. The function `startingPoints.m` returns the number of starting points available for one specific test problem by accepting a string of characters, corresponding to the test problem name, as the only input argument. Finally the function `startingPoints.m` returns a starting point for a test problem by accepting in input two arguments: a string of characters corresponding to the test problem name and a positive integer which selects the desired starting point of such problem. Table 5 summarizes all possible utilizations of the function `startingPoints`.

3 Test Problem Descriptions

In this section we report the test problems. The section is divided into three subsections. Subsection 3.1 contains pure academic problems, subsection 3.2 contains QVIs modelling some real problems, while subsection 3.3 contains discretization of infinite dimensional problems.

`startingPoints`	displays a list of all test problems available with a brief description of their starting points
`startingPoints(QVIname)`	returns the number of starting points available for the test problem `QVIname` (the input argument (`QVIname` must be a string of characters)
`startingPoints(QVIname,number)`	returns the `number`-th starting point for the test problem `QVIname` (the input argument `QVIname` must be a string of characters, while the input argument `number` must be an integer)

Table 5: Possible utilizations of the M-file `startingPoints.m`

3.1 Academic Problems

OutZ40 [LBB-A-2-4-0-2-0]

source: [13, p. 13]

description:

$$\begin{aligned} F(x) &:= \begin{pmatrix} 2 & 8/3 \\ 5/4 & 2 \end{pmatrix} x - \begin{pmatrix} 34 \\ 24.25 \end{pmatrix}, \\ g^I(y) &:= \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} y - \begin{pmatrix} 0 \\ 11 \\ 0 \\ 11 \end{pmatrix}, \\ g^P(y,x) &:= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} y + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} x - \begin{pmatrix} 15 \\ 15 \end{pmatrix} \end{aligned}$$

JF: positive definite (everywhere)

comments: this problem was built so that it does not satisfy the assumptions for the local convergence of the Newton method discussed in [13] at the known solution listed below

known solution: $x^* = (10, 5)^T$

OutZ41 [LBB-A-2-4-0-2-0]

source: [13, Example 4.1]

description:

$$F(x) := \begin{pmatrix} 2 & 8/3 \\ 5/4 & 2 \end{pmatrix} x - \begin{pmatrix} 100/3 \\ 22.5 \end{pmatrix},$$

$$g^I(y) := \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} y - \begin{pmatrix} 0 \\ 11 \\ 0 \\ 11 \end{pmatrix},$$

$$g^P(y,x) := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} y + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} x - \begin{pmatrix} 15 \\ 20 \end{pmatrix}$$

JF: positive definite (everywhere)

comments: a variant of the OutZ40 that satisfies the assumptions for the local convergence of the Newton method discussed in [13] at the known solution listed below

known solution: $x^* = (10, 5)^T$

OutZ42 [LBB-A-4-4-0-4-0]

source: [13, Example 4.2]

description:

$$F(x) := \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$g^I(y) := y,$$

$$g^P(y,x) := \begin{pmatrix} -y_1 - 2.5 + x_1 + x_1^2 \\ \vdots \\ -y_4 - 2.5 + x_4 + x_4^2 \end{pmatrix}$$

JF: positive definite (everywhere)

known solution: $x^* \approx (-1.291, -1.5811, -1.5811, -1.291)^T$

OutZ43 [LAB-A-4-0-0-4-0]

source: [13, Example 4.3]

description:

$$F(x) := \text{same as for problem OutZ42},$$

$$g^P(y,x) := -y - \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} x - \begin{pmatrix} 1.5 \\ 1.5 \\ 1.5 \\ 1.5 \end{pmatrix}$$

JF: positive definite (everywhere)

comments: this problem satisfies conditions of Theorem 7 in [4]

known solution: $x^* = (-0.9, -1.2, -1.2, -0.9)^T$

OutZ44 [LAB-A-4-0-0-4-0]

source: [13, Example 4.4]

description:

$$\begin{array}{rcl} F(x) & := & \text{same as for problem OutZ42} \\ g^P(y,x) & := & -y - 1.5 \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} x - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \\ & & +0.25 \begin{pmatrix} (2x_1 - x_2 + 1)^2 \\ (-x_1 + 2x_2 - x_3 + 1)^2 \\ (-x_2 + 2x_3 - x_4 + 1)^2 \\ (-x_3 + 2x_4 + 1)^2 \end{pmatrix} \end{array}$$

JF: positive definite (everywhere)

known solution: $x^* \approx (-1.0021, -1.36, -1.36, -1.0021)^T$

Moving set problems

This is the most studied class of QVIs, namely problems where the set $\tilde{K}(x)$ is given by

$$\tilde{K}(x) := c(x) + Q$$

for some function $c : \mathbb{R}^n \to \mathbb{R}^n$ and a fixed set $Q \subseteq \mathbb{R}^n$. Assuming that Q has a representation of the form

$$Q := \{x \in \mathbb{R}^n \mid q(x) \leq 0\},$$

for some function $q : \mathbb{R}^n \to \mathbb{R}^{m_P}$, it is easy to see that $\tilde{K}(x)$ can be rewritten as

$$\tilde{K}(x) := \{y \in \mathbb{R}^n \mid q(y - c(x)) \leq 0\}$$

which corresponds to the general setting considered in this paper of a QVI with $g^P : \mathbb{R}^n \to \mathbb{R}^{m_P}$ being defined by

$$g^P(y,x) := q(y - c(x)). \qquad (2)$$

MovSet1A - MovSet1B [LAO-A-5-0-0-1-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= \|y - \alpha x\|^2 - 0.5 \end{aligned}$$

with

$$A := \begin{pmatrix} 19.8699 & 0.5369 & 2.9482 & 0.3358 & 7.1239 \\ 4.1819 & 16.3484 & -5.2030 & 5.4332 & 2.7143 \\ -5.6554 & 0.9422 & 19.0981 & 7.1556 & -7.3810 \\ -1.8770 & 0.1918 & -5.3596 & 18.3565 & -7.8847 \\ -6.0303 & -3.6171 & -1.4658 & 4.6238 & 15.4085 \end{pmatrix}, \quad b := \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 10 \end{pmatrix}, \tag{3}$$

and $\alpha := 0.1$ for MovSet1A and $\alpha := 10$ for MovSet1B

JF: positive definite (everywhere)

comments: referring to the general description (2) of g^P: $q(z) := \|z\|^2 - 0.5$, $c(x) := \alpha x$. Note that MovSet1A satisfies conditions of Theorem 4 in [4], while MovSet1B does not

MovSet2A - MovSet2B [LAO-A-5-0-0-1-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= \|y - \alpha\big(\cos(x_i)\big)_{i=1}^n\|^2 - 0.5 \end{aligned}$$

with A and b as in (3) and $\alpha := 0.1$ for MovSet2A and $\alpha := 10$ for MovSet2B

JF: positive definite (everywhere)

comments: referring to the general description (2) of g^P: $q(z) := \|z\|^2 - 0.5$, $c(x) := \alpha\big(\cos(x_i)\big)_{i=1}^n$. Note that MovSet2A satisfies conditions of Theorem 4 in [4], while MovSet2B does not

MovSet3A1 - MovSet3B1 [LAO-A-1000-0-0-1-0]

MovSet3A2 - MovSet3B2 [LAO-A-2000-0-0-1-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= (y - Mx)^T Q(y - Mx) - d \end{aligned}$$

where A, b, M, Q and d are available in the corresponding Matlab functions (MovSet3A1 and MovSet3A2 differ from MovSet3B1 and MovSet3B2, respectively, only in the matrix M)

JF: positive definite (everywhere)

comments: referring to the general description (2) of g^P: $q(z) := z^T Qz - d$, $c(x) := Mx$. Note that MovSet3A1 and MovSet3A2 satisfy conditions of Theorem 4 in [4], while MovSet3B1 and MovSet3B2 do not

MovSet4A1 - MovSet4B1 [LAO-A-400-0-0-801-0]

MovSet4A2 - MovSet4B2 [LAO-A-800-0-0-1601-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= \begin{pmatrix} -y + Mx \\ y - Mx - 1 \\ \mathbf{1}_n^T y - \mathbf{1}_n^T Mx - \frac{n}{2} \end{pmatrix} \end{aligned}$$

where A, b and M are available in the corresponding Matlab functions (MovSet4A1 and MovSet4A2 differ from MovSet4B1 and MovSet4B2, respectively, only in the matrix M)

JF: positive definite (everywhere)

comments: referring to the general description (2) of g^P:

$$q(z) := \begin{pmatrix} -z \\ z - 1 \\ \mathbf{1}_n^T z - \frac{n}{2} \end{pmatrix}, \quad c(x) := Mx.$$

Note that MovSet4A1 and MovSet4A2 satisfy conditions of Theorem 4 in [4], while MovSet4B1 and MovSet4B2 do not

Problems with box constraints

This class of QVIs have a set $\tilde{K}(x)$ defined by constraints of the form

$$g^P(y,x) := \begin{pmatrix} (y_i - s_i x_i - c_i)_{i=1}^n \\ (-y_i + t_i x_i - d_i)_{i=1}^n \end{pmatrix}. \tag{4}$$

We call this a QVI with box constraints since, given a fixed vector x, the feasible set describes box constraints for the variables y. The particular values of the box constraints for a variable y_i, however, varies with x_i.

Box1A - Box1B [LAB-A-5-0-0-10-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= \begin{pmatrix} (y_i - \alpha x_i - c_i)_{i=1}^n \\ (-y_i + \alpha x_i - c_i)_{i=1}^n \end{pmatrix} \end{aligned}$$

where A and b are defined as in (3),

$$c := \begin{pmatrix} 0.1202 \\ 1.7418 \\ 2.7064 \\ 2.0502 \\ 4.4616 \end{pmatrix}$$

and $\alpha := 0.1$ for Box1A and $\alpha := 2$ for Box1B

JF: positive definite (everywhere)

comments: Box1A satisfies conditions of Corollary 6 in [4], while Box1B does not

Box2A - Box2B [NAB-A-500-0-0-1000-0]

source: this paper

description:

$$F(x) := Ax + b + (\exp(x_i))_{i=1}^{500},$$

g^P is defined as in (4), where A, b, s, t, c and d are available in the corresponding Matlab functions (Box2A differs from Box2B only in the vectors s and t)

JF: positive definite (everywhere)

comments: Box2A satisfies conditions of Corollary 6 in [4], while Box2B does not

Box3A - Box3B [NAB-A-500-0-0-1000-0]

source: this paper

description: these problems are identical to Box2 except the function F:

$$F(x) := Ax + b + M\left(x_i^3\right)_{i=1}^{500},$$

A, b, M, s, t, c and d are available in the corresponding Matlab functions (Box3A differs from Box3B only in the vectors s and t)

JF: positive definite (everywhere)

comments: Box3A satisfies conditions of Corollary 6 in [4], while Box3B does not

Problems with bilinear constraints

In these problems, the set $\tilde{K}(x)$ is defined by the following inequality constraints only:

$$g^P(y,x) := \begin{pmatrix} x^T Q_1 y - c_1 \\ \vdots \\ x^T Q_q y - c_q \end{pmatrix}.$$

BiLin1A - BiLin1B [LBO-A-5-10-0-3-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^I(y) &:= \begin{pmatrix} l - y \\ y - u \end{pmatrix}, \\ g^P(y,x) &:= \begin{pmatrix} x^T Q_1 y - c_1 \\ \vdots \\ x^T Q_3 y - c_3 \end{pmatrix}, \end{aligned}$$

where A and b are defined as in (3),

$$l := \begin{pmatrix} -0.1202 \\ -1.7418 \\ -2.7064 \\ -2.0502 \\ -4.4616 \end{pmatrix}, \quad u := -l, \quad c := \begin{pmatrix} 0.3070 \\ 1.1186 \\ 2.6149 \end{pmatrix},$$

$$Q_1 := \begin{pmatrix} 1.9073 & 0.2403 & 0.2352 & -0.4903 & -0.2651 \\ 0.2403 & 1.1319 & 1.2087 & -0.3268 & 0.2540 \\ 0.2352 & 1.2087 & 1.6862 & 0.2941 & 0.6732 \\ -0.4903 & -0.3268 & 0.2941 & 1.8258 & 0.1363 \\ -0.2651 & 0.2540 & 0.6732 & 0.1363 & 1.5527 \end{pmatrix} + \alpha \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$Q_2 := \begin{pmatrix} 2.7307 & 0.5988 & 1.5728 & 1.4072 & -0.3082 \\ 0.5988 & 2.2435 & 0.7546 & 1.3632 & 1.5852 \\ 1.5728 & 0.7546 & 2.3809 & 1.2625 & 1.0403 \\ 1.4072 & 1.3632 & 1.2625 & 1.7612 & 0.3071 \\ -0.3082 & 1.5852 & 1.0403 & 0.3071 & 2.6305 \end{pmatrix} + \alpha \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$Q_3 := \begin{pmatrix} 2.5189 & 2.1947 & 1.7697 & 2.2753 & 1.9885 \\ 2.1947 & 3.8143 & 1.3839 & 1.5636 & 1.8451 \\ 1.7697 & 1.3839 & 3.3655 & 1.6441 & 1.9946 \\ 2.2753 & 1.5636 & 1.6441 & 3.6885 & 2.3272 \\ 1.9885 & 1.8451 & 1.9946 & 2.3272 & 2.2883 \end{pmatrix} + \alpha \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and $\alpha := 0$ for BiLin1A and $\alpha := 10$ for BiLin1B

JF: positive definite (everywhere)

comments: BiLin1A satisfies conditions of Corollary 7 in [4], while BiLin1B does not

Problems with linear constraints with variable right-hand side

In this class of problems, the feasible set $\tilde{K}(x)$ is defined by

$$g^P(y,x) := Ey - d + c(x),$$

where $E \in \mathbb{R}^{m \times n}$ is a given matrix, $c : \mathbb{R}^n \to \mathbb{R}^{m_P}$ and $d \in \mathbb{R}^{m_P}$. In this class of QVIs, the feasible set is defined by linear inequalities in which the right-hand side depends on x.

RHS1A1 - RHS1B1 [LAL-A-200-0-0-199-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= Ey - d + C\,(\sin(x_i))_{i=1}^n \end{aligned}$$

where A, b, E, d and C are available in the corresponding Matlab functions (RHS1A1 differs from RHS1B1 only in the matrix C)

JF: positive definite (everywhere)

comments: RHS1A1 satisfies conditions of Theorem 5 in [4], while RHS1B1 does not

RHS2A1 - RHS2B1 [LAL-A-200-0-0-199-0]

source: this paper

description:

$$\begin{aligned} F(x) &:= Ax + b, \\ g^P(y,x) &:= Ey - d + Cx \end{aligned}$$

where A, b, E, d and C are available in the corresponding Matlab functions (RHS2A1 differs from RHS2B1 only in the matrix C)

JF: positive definite (everywhere)

comments: RHS2A1 satisfies conditions of Theorem 5 in [4], while RHS2B1 does not

3.2 Problems from Real-World Models

Walrasian equilibrium problems

Problems in this subsection are QVI reformulations of a Walrasian pure exchange economy with utility function without production whose general structure is described [1]; the specific data used here are taken from [3]. In this model there are C agents, whose preferences are given by a utility function u_i, exchanging P goods. Each agent controls a variable $x^i \in \mathbb{R}^P$ (representing quantity of goods) and has an initial endowment of $\xi^i \in \mathbb{R}^P$. There is also one extra, 0-th "player" controlling a vector $x^0 \in \mathbb{R}^P$ representing prices. Therefore, the vector of variables is $x = (x^i)_{i=0}^C \in \mathbb{R}^{(C+1)P}$. The dimensions and the description of the QVI model depend on the parameters C and P:

$$
\begin{aligned}
n &:= P(C+1), \quad m_I := P(C+1), \quad p_I := 1, \quad m_P := C, \quad p_P := 0, \\
F(x) &:= \begin{pmatrix} \sum_{i=1}^C \xi^i - x^i \\ \nabla_{x^1} u_1(x^1) \\ \vdots \\ \nabla_{x^C} u_C(x^C) \end{pmatrix}, \quad g^I(y) := -y, \quad M^I := (\mathbf{1}_P^T \quad \mathbf{0}_{PC}^T), \qquad (5) \\
v^I &:= -1, \quad g^P(y,x) := \begin{pmatrix} \sum_{j=1}^P x_j^0 (y_j^1 - \xi_j^1) \\ \vdots \\ \sum_{j=1}^P x_j^0 (y_j^C - \xi_j^C) \end{pmatrix}.
\end{aligned}
$$

The utility functions u of the agents, as well as the parameters C and P and the endowment ξ, are specified for each test problem.

WalEq1 [LBO-R-18-18-1-5-0]

source: model from [1], data from [3]

description: the general description is (5) where $C := 5$, $P := 3$, the utility functions are quadratic and convex:

$$
\begin{aligned}
u_i(x^i) &:= \frac{1}{2}(x^i)^T Q^i x^i - (b^i)^T x^i, \quad i = 1, \ldots, 5, \\
Q^i &:= \begin{pmatrix} 6 & -2 & 5 \\ -2 & 6 & -7 \\ 5 & -7 & 20 \end{pmatrix}, \quad b^i := \begin{pmatrix} 32+i \\ 32+i \\ 32+i \end{pmatrix}, \quad i = 1, 2, \\
Q^i &:= \begin{pmatrix} 6 & 1 & 0 \\ 1 & 7 & -5 \\ 0 & -5 & 7 \end{pmatrix}, \quad b^i := \begin{pmatrix} 30+(i+2)*2 \\ 30+(i+2)*2 \\ 30+(i+2)*2 \end{pmatrix}, \quad i = 3, 4, 5,
\end{aligned}
$$

and

$$
\xi^i := \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}, \quad i = 1, 2, \qquad \xi^i := \begin{pmatrix} 6 \\ 5 \\ 4 \end{pmatrix}, \quad i = 3, 4, 5
$$

JF: P_0 (everywhere) but never positive semidefinite

WalEq2 [NBO-R-105-105-1-20-0]

source: model from [1], data from [3]

description: the general description is (5) where $C := 20$, $P := 5$, the utility functions are of logarithmic type:

$$u_i(x^i) := -\sum_{k=1}^{5}(a_k + i + 4)\log(x_k^i + b_k + 2(i+4)), \qquad i = 1,\dots,10,$$

$$u_i(x^i) := -\sum_{k=1}^{5}(c_k + i + 4)\log(x_k^i + d_k + i + 4), \qquad i = 11,\dots,20,$$

$$a := \begin{pmatrix}1\\2\\4\\6\\8\end{pmatrix}, \quad b := \begin{pmatrix}20\\30\\30\\40\\50\end{pmatrix}, \quad c := \begin{pmatrix}10\\6\\4\\10\\1\end{pmatrix}, \quad d := \begin{pmatrix}50\\40\\30\\20\\20\end{pmatrix},$$

and

$$\xi^i := \begin{pmatrix}2\\3\\4\\1\\6\end{pmatrix}, \quad i = 1,\dots,10, \qquad \xi^i := \begin{pmatrix}6\\5\\4\\3\\2\end{pmatrix}, \quad i = 11,\dots,20$$

JF: P_0 (everywhere) but never positive semidefinite

WalEq3 [LBO-R-186-186-1-30-0]

source: model from [1], data from [3]

description: the general description is (5) where $C := 30$, $P := 6$, the utility functions are quadratic and convex:

$$u_i(x^i) := \frac{1}{2}(x^i)^T Q^i x^i - (b^i)^T x^i, \quad i = 1,\dots,30,$$

$$Q^i := A,\ b^i := \begin{pmatrix}56+i\\66+i\\76+i\\66+i\\66+i\\56+i\end{pmatrix},\ i = 1,\dots,15,$$

$$Q^i := B, \quad b^i := \begin{pmatrix}50+2*(i+6)\\60+2*(i+6)\\50+2*(i+6)\\70+2*(i+6)\\70+2*(i+6)\\60+2*(i+6)\end{pmatrix},\ i = 16,\dots,30,$$

and

$$\xi^i := \begin{pmatrix}2\\3\\4\\1\\6\\1\end{pmatrix}, \quad i = 1,\dots,15, \qquad \xi^i := \begin{pmatrix}6\\5\\4\\3\\2\\8\end{pmatrix}, \quad i = 16,\dots,30.$$

Set A equal to

```
 68.22249416536778  12.12481199690621  -8.35496210217478  -6.81177486915109  -4.66752803051747  3.64100170417482
 12.12481199690621  53.51450780426463 -21.77618227261339 -15.00376305863444  -0.11788350473544  2.03354709400720
 -8.35496210217478 -21.77618227261339  35.44033408387684   4.35160649036518  19.17472558234163 -3.40090742729160
 -6.81177486915109 -15.00376305863444   4.35160649036518  52.25155022199242  -5.99490328518247 20.40443259092577
 -4.66752803051747  -0.11788350473544  19.17472558234163  -5.99490328518247  23.32798561358070 -3.58535668529727
  3.64100170417482   2.03354709400720  -3.40090742729160  20.40443259092577  -3.58535668529727 10.21258119890765
```

and B equal to

```
 61.74633559943146 -23.83006225091380  16.78581949473039  14.42073900860500  -2.75188745616575  13.44307656650567
-23.83006225091380  37.64246654306209  -3.76510322128227  16.32022449045404 -39.90743633716275  11.38657250296817
 16.78581949473039  -3.76510322128227  53.34843665848310   4.60388415537161 -23.04611587657949 -25.31392346426841
 14.42073900860500  16.32022449045404   4.60388415537161  40.69699687713468 -30.78019133996427  17.08866411420883
 -2.75188745616575 -39.90743633716275 -23.04611587657949 -30.78019133996427  66.22678445157413 -12.28091080313848
 13.44307656650567  11.38657250296817 -25.31392346426841  17.08866411420883 -12.28091080313848  41.37849544246254
```

JF: P_0 (everywhere) but never positive semidefinite

WalEq4 [NBO-R-310-310-1-30-0]

source: model from [1], data from [3]

description: the general description is (5) where $C := 30$, $P := 10$, the utility functions are of logarithmic type:

$$u_i(x^i) \;:=\; -\sum_{k=1}^{10}(a_k + i + 6)\log(x_k^i + b_k + 2(i+6)), \qquad i \;=\; 1,\ldots,15,$$

$$u_i(x^i) \;:=\; -\sum_{k=1}^{10}(c_k + i + 6)\log(x_k^i + d_k + i + 6), \qquad i \;=\; 16,\ldots,30,$$

$$a \;:=\; \begin{pmatrix} 1\\2\\4\\6\\8\\7\\8\\10\\1\\5 \end{pmatrix}, \quad b := \begin{pmatrix} 50\\60\\70\\60\\60\\50\\50\\80\\60\\70 \end{pmatrix}, \quad c := \begin{pmatrix} 10\\6\\4\\10\\1\\2\\6\\4\\9\\4 \end{pmatrix}, \quad d := \begin{pmatrix} 50\\60\\50\\70\\70\\60\\50\\50\\80\\50 \end{pmatrix},$$

and

$$\xi^i \;:=\; \begin{pmatrix} 2\\3\\4\\1\\6\\1\\3\\6\\2\\10 \end{pmatrix}, \quad i \;=\; 1,\ldots,15, \qquad \xi^i := \begin{pmatrix} 6\\5\\4\\3\\2\\8\\4\\6\\2\\0 \end{pmatrix}, \quad i \;=\; 16,\ldots,30$$

JF: P_0 (everywhere) but never positive semidefinite

WalEq5 [NBO-R-492-492-1-40-0]

source: model from [1], data from [3]

description: the general description is (5) where $C := 40$, $P := 12$, the utility functions are of logarithmic type:

$$u_i(x^i) := -\sum_{k=1}^{12} (a_k + i + 7) \log(x_k^i + b_k + 2(i+7)), \qquad i = 1, \ldots, 20,$$

$$u_i(x^i) := -\sum_{k=1}^{12} (c_k + i + 7) \log(x_k^i + d_k + i + 7), \qquad i = 21, \ldots, 40,$$

$$a := \begin{pmatrix} 1 \\ 2 \\ 4 \\ 6 \\ 8 \\ 7 \\ 8 \\ 10 \\ 1 \\ 5 \\ 2 \\ 4 \end{pmatrix}, \quad b := \begin{pmatrix} 50 \\ 60 \\ 70 \\ 60 \\ 60 \\ 50 \\ 50 \\ 80 \\ 60 \\ 70 \\ 70 \\ 80 \end{pmatrix}, \quad c := \begin{pmatrix} 10 \\ 6 \\ 4 \\ 10 \\ 1 \\ 2 \\ 6 \\ 4 \\ 9 \\ 4 \\ 5 \\ 1 \end{pmatrix}, \quad d := \begin{pmatrix} 50 \\ 60 \\ 50 \\ 70 \\ 70 \\ 60 \\ 50 \\ 50 \\ 80 \\ 50 \\ 60 \\ 70 \end{pmatrix},$$

and

$$\xi^i := \begin{pmatrix} 2 \\ 3 \\ 4 \\ 1 \\ 6 \\ 1 \\ 3 \\ 6 \\ 2 \\ 10 \\ 3 \\ 4 \end{pmatrix}, \quad i = 1, \ldots, 20, \qquad \xi^i := \begin{pmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 8 \\ 4 \\ 6 \\ 2 \\ 0 \\ 6 \\ 0 \end{pmatrix}, \quad i = 21, \ldots, 40$$

JF: P_0 (everywhere) but never positive semidefinite

Wal2 [NLO-A-105-107-0-20-0]

source: model from [1], data from [3]

description: the general description is (5), except for

$$m_I := P(C+1) + 2, \quad p_I := 0,$$

$$g^I(y) := \begin{pmatrix} -y \\ \sum_{i=0}^{P} y_i^0 - 1 \\ 1 - \sum_{i=0}^{P} y_i^0 \end{pmatrix},$$

where $C := 20$, $P := 5$ and the utility functions are of logarithmic type:

$$u_i(x^i) := \sum_{k=1}^{5} (a_k + k + 4) \log(x_k^i + b_k + 2(i+4)), \qquad i = 1, \ldots, 10,$$

$$u_i(x^i) \;:=\; \sum_{k=1}^{5}(c_k+k+4)\log(x^i_k+d_k+i+4), \qquad i \;=\; 11,\ldots,20,$$

a, b, c, d and ξ are the same as for WalEq2

JF: never P_0

comments: this QVI arises from an implementation mistake, in fact it differs from WalEq2 essentially only in the sign and in one parameter of the u_i functions; furthermore the equality constraint $\sum_{i=1}^{P} y_i^0 = 1$ is substituted by a double inequality. Since the problem proved challenging, we kept it in the library.

Wal3 [LLO-R-186-188-0-30-0]

source: model from [1], data from [3]

description: the general description is (5), except for

$$m_I \;:=\; P(C+1)+2, \quad p_I := 0,$$

$$g^I(y) \;:=\; \begin{pmatrix} -y \\ \sum_{i=0}^{P} y_i^0 - 1 \\ 1-\sum_{i=0}^{P} y_i^0 \end{pmatrix},$$

where all parameters and functions are the same as for WalEq3

JF: P_0 (everywhere) but never positive semidefinite

comments: this QVI differs from WalEq3 only for the fact that the equality constraint $\sum_{i=1}^{P} y_i^0 = 1$ is substituted by a double inequality.

Wal5 [NLO-A-492-494-0-40-0]

source: model from [1], data from [3]

description: the general description is (5), except for

$$m_I \;:=\; P(C+1)+2, \quad p_I := 0,$$

$$g^I(y) \;:=\; \begin{pmatrix} -y \\ \sum_{i=0}^{P} y_i^0 - 1 \\ 1-\sum_{i=0}^{P} y_i^0 \end{pmatrix},$$

where $C := 40$, $P := 12$, the utility functions are of logarithmic type:

$$u_i(x^i) \;:=\; \sum_{k=1}^{12}(a_k+k+7)\log(x^i_k+b_k+2(i+7)), \qquad i \;=\; 1,\ldots,20,$$

$$u_i(x^i) \;:=\; \sum_{k=1}^{12}(c_k + k + 7)\log(x_k^i + d_k + i + 7), \qquad i \;=\; 21, \ldots, 40,$$

a, b, c, d and ξ are the same as for WalEq5

JF: never P_0

comments: this QVI arises from an implementation mistake, in fact it differs from WalEq2 essentially only in the sign and in one parameter of the u_i functions; furthermore the equality constraint $\sum_{i=1}^{P} y_i^0 \;=\; 1$ is substituted by a double inequality. Since the problem proved challenging, we kept it in the library.

Generalized Nash equilibrium problems

It is well known that finding an equilibrium of a generalized Nash game is equivalent to solving a QVI problem, see [2]. This QVI model of an energy market Nash equilibrium is taken from [9]. Let N agents owns l plants each one to generate electric energy for sale. We denote as x_j^i the energy produced by agent i in the j-th plant. The unitary energy price in the market depends on the total amount of energy produced by all agents, it is modelled by a quadratic concave function of one variable. Then the profit of each agent depends on the generation level of the other agents in the market. In turn, each generation level is constrained by technological limitations of the power plants. The coordination, or regulation, of the market is done by the Independent System Operator (ISO), whose actions in the market are considered as those of an additional player. Accordingly, letting the ISO be player number 0, the ISO tries to maximize the social welfare by encouraging all other agents to satisfy the total market demand.

$$n \;:=\; Nl + 1, \quad m_I := 2(Nl + 1), \quad p_P := N + 1,$$

$$F(x) \;:=\; \begin{bmatrix} 0 \\ c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)^2 - 120 + 2c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)\left(\sum_{j=1}^l x_j^1\right) \\ \vdots \\ c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)^2 - 120 + 2c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)\left(\sum_{j=1}^l x_j^1\right) \\ \vdots \\ c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)^2 - 120 + 2c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)\left(\sum_{j=1}^l x_j^N\right) \\ \vdots \\ c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)^2 - 120 + 2c\left(\sum_{i=1}^N \sum_{j=1}^l x_j^i\right)\left(\sum_{j=1}^l x_j^N\right) \end{bmatrix} + \begin{bmatrix} 0 \\ A_1^1 x_1^1 \\ \vdots \\ A_l^1 x_l^1 \\ \vdots \\ A_1^N x_1^N \\ \vdots \\ A_l^N x_l^N \end{bmatrix} + \begin{bmatrix} 120 \\ b_1^1 \\ \vdots \\ b_l^1 \\ \vdots \\ b_1^N \\ \vdots \\ b_l^N \end{bmatrix}, \tag{6}$$

$$g^I(y) \;:=\; \begin{bmatrix} -y^0 \\ -y^1 \\ \vdots \\ -y^N \\ y^0 - u^0 \\ y^1 - u^1 \\ \vdots \\ y^N - u^N \end{bmatrix}, \qquad M^P(x) := \begin{pmatrix} 1 & & & \\ & \mathbf{1}_l^T & & \\ & & \ddots & \\ & & & \mathbf{1}_l^T \end{pmatrix},$$

$$v^P(x) \;:=\; \begin{pmatrix} 0 & \mathbf{1}_l^T & & \mathbf{1}_l^T \\ \mathbf{1}_l^T & \mathbf{0}_l^T & & \mathbf{1}_l^T \\ & & \ddots & \\ \mathbf{1}_l^T & \mathbf{1}_l^T & & \mathbf{0}_l^T \end{pmatrix} \begin{pmatrix} x^0 \\ x^1 \\ \vdots \\ x^N \end{pmatrix} - \begin{pmatrix} d \\ d \\ \vdots \\ d \end{pmatrix},$$

where $c \in \mathbb{R}, A \in \mathbb{R}^{Nl}, b^{Nl}, u \in \mathbb{R}^{Nl}$ and $d \in \mathbb{R}$.

LunSS1 [NBA-R-501-1002-0-0-6]

LunSS2 [NBA-R-1251-2502-0-0-6]

LunSS3 [NBA-R-5001-10002-0-0-6]

source: [9]

description: these problems are described by (6), where $N := 5$ and $l := 100$ for LunSS1, $l := 250$ for LunSS2, $l := 1000$ for LunSS3. c, A, b, u and d are available for all these problems in the corresponding Matlab functions.

It is possible to compute some equilibria (not all in general) of a jointly convex Nash problem by solving a variational inequality instead. These points are called variational equilibria and have some properties from the economic point of view, see [2]. The next 3 problems pursue this goal. Note that those problems are pure VIs in which the parametric set $\tilde{K}(x)$ vanishes and $K(x) = \bar{K}$.

LunSSVI1 [NBA-R-501-1002-1-0-0]

LunSSVI2 [NBA-R-1251-2502-1-0-0]

LunSSVI3 [NBA-R-5001-10002-1-0-0]

source: [9]

description: F and g^I are taken from (6) while $M^I := \mathbf{1}_n^T$, and $v^I := -d$, where $N := 5$ and $l := 100$ for LunSSVI1, $l := 250$ for LunSSVI2, $l := 1000$ for LunSSVI3. c, A, b, u and d are available for all these problems in the corresponding Matlab functions. Note that $p_P := 0$.

comments: these problems are pure VIs

3.3 Discretized Problems

In this section consider finite dimensional QVIs obtained by making a discretization procedure on infinite dimensional QVIs. This series of problems stemmed from different fields.

Transportation problems

In the modeling of competition on networks in [16] it is assumed that users either behave following the Wardrop equilibrium or the Nash equilibrium concept. In the time-dependent network model shared by two types of users: group users (Nash players) and individual users (Wardrop players), both classes of users choose the paths to ship their jobs so as to minimize their costs, but they apply different optimization criteria. The source of interaction of users is represented by the travel demand, which is assumed to be elastic with respect to the equilibrium solution. Thus, the equilibrium distribution is proved to be equivalent to the solution of an appropriate time-dependent quasi-variational inequality problem. This example taken from [16] is relative to a simple network with 4 nodes and 7 edges, in which there are two users: one Nash user and one Wardrop user. The time interval considered is $[0, N]$, and in particular it is discretized so that the time instants are $0, \dots, N$. Then the solution of the following QVI contains the flows on the paths of the network at the equilibrium in the instants $1, \dots, N$ for the two users. The dimensions and the description of the following two discretized models depend on the parameter N:

$$
\begin{aligned}
n &:= 4N, \quad m^I := 4N, \quad p^P := 2N, \\
F(x) &:= \begin{pmatrix} A & & 0 \\ & \ddots & \\ 0 & & A \end{pmatrix} x + \begin{pmatrix} b \\ \vdots \\ b \end{pmatrix}, \\
g^I(y) &:= -y, \quad M^P(x) := \begin{pmatrix} C & & 0 \\ & \ddots & \\ 0 & & C \end{pmatrix}, \\
v^P(x) &:= \begin{pmatrix} -E & & 0 \\ & \ddots & \\ 0 & & -E \end{pmatrix} x + \begin{pmatrix} -d_1 - d_2(1-1)/(N-1) \\ \vdots \\ -d_1 - d_2(N-1)/(N-1) \end{pmatrix},
\end{aligned}
\tag{7}
$$

where

$$
A := \begin{pmatrix} 4 & 2 & 0 & 0 \\ 2 & 10 & 0 & 4 \\ 0 & 0 & 2 & 1 \\ 0 & 2 & 1 & 5 \end{pmatrix}, \quad b := \begin{pmatrix} 40 \\ 30 \\ 40 \\ 30 \end{pmatrix}, \quad C := \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix},
$$

$$
E := \begin{pmatrix} 2/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}, \quad d1 := \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad d2 := \begin{pmatrix} 10 \\ 4 \end{pmatrix}
$$

Scrim11 [LBA-D-2400-2400-0-0-1200]

Scrim12 [LBA-D-4800-4800-0-0-2400]

source: [16]

description: the general description is (7) with $N := 600$ for Scrim11 and $N := 1200$ for Scrim12

JF: positive definite (everywhere)

Scrim21 [LBL-D-2400-2400-0-2400-0]

Scrim22 [LBL-D-4800-4800-0-4800-0]

source: [16]

description: the general description is (7), but here

$$m^P := 4N, \quad p^P := 0,$$

$$g^P(y,x) := \begin{pmatrix} C & & 0 \\ & \ddots & \\ 0 & & C \\ -C & & 0 \\ & \ddots & \\ 0 & & -C \end{pmatrix} y + \begin{pmatrix} -E & & 0 \\ & \ddots & \\ 0 & & -E \\ E & & 0 \\ & \ddots & \\ 0 & & E \end{pmatrix} x + \begin{pmatrix} -d_1 - d_2(1-1)/(N-1) \\ \vdots \\ -d_1 - d_2(N-1)/(N-1) \\ d_1 + d_2(1-1)/(N-1) \\ \vdots \\ d_1 + d_2(N-1)/(N-1) \end{pmatrix},$$

with $N := 600$ for Scrim21 and $N := 1200$ for Scrim22

JF: positive definite (everywhere)

comments: Problems Scrim21 and Scrim22 are essentially the same as problems Scrim11 and Scrim22, respectively, except that each equality constraint has been rewritten as two inequalities. In particular, the standard linear independence constraint qualification is therefore violated for problems Scrim21 and Scrim22.

Contact problems with Coulomb friction

This is the problem of an elastic body in contrast to a rigid obstacle. In particular, this is the most realistic model in which Coulomb friction is present (in this problem $\phi \in \mathbb{R}$ is the friction coefficient). The problem is taken from Example 11.1 in [12]. Let $x^* \in \mathbb{R}^n$ be a solution of the QVI, then odd elements of x^* are interpreted as tangential stress components on the rigid obstacle in different points of such obstacle, while even elements are interpreted as outer normal ones. We consider different instances of this problem which derive from different discretizations generating different fragmentations of the obstacle in identical segments. In particular, the case in which the obstacle is divided into N segments involves $2(N+1)$ variables in the model (since there are $N+1$ extreme segment points and having to consider both tangential and outer normal stress components for all of them).

The dimensions and the description of the following two discretized models depend on the parameter N:

$$
\begin{aligned}
n &:= 2(N+1), \quad m^I := 2(N+1), \quad m^P := 2(N+1), \\
F(x) &:= Ax - b, \quad g^I(y) := \begin{bmatrix} (-y_{2i} - 10)_{i=1}^{N+1} \\ (y_{2i})_{i=1}^{N+1} \end{bmatrix}, \\
g^P(y,x) &:= \begin{bmatrix} \left(-y_{(2i-1)} - \phi x_{2i}\right)_{i=1}^{N+1} \\ \left(y_{(2i-1)} - \phi x_{2i}\right)_{i=1}^{N+1} \end{bmatrix},
\end{aligned} \tag{8}
$$

where the positive definite square matrix A and the vector b depend on N and are available in the library for $N := 30$ and $N := 40$ (data for these problems have been kindly provided by J.V. Outrata, M. Kočvara and J. Zowe).

OutKZ31 [LBB-D-62-62-0-62-0]

OutKZ41 [LBB-D-82-82-0-82-0]

source: [12]

description: the general description is (8) with friction coefficient $\phi := 10$ and the fragmentation granularity $N := 30$ for OutKZ31 and $N := 40$ for OutKZ41

JF: positive definite (everywhere)

QVIs with gradient constraints

The problems considered here are taken from [8] (see also [7]) and represent a stationary model for the magnetization of type-II superconductors.

Specifically, let $\Omega \subseteq \mathbb{R}^2$ be an open and convex domain, let $W := W^{1,2}(\Omega)$ be the corresponding Sobolev space, and let j_c be a nonnegative continuous function. Then the infinite-dimensional problem from [8] (using $p = 2$) is to find a solution $u \in K(u)$ satisfying

$$\int_\Omega \nabla u(\xi)^T \nabla(v-u)(\xi) d\xi \geq 0 \quad \forall v \in K(u), \tag{9}$$

where the feasible set $K(u)$ is defined by

$$K(u) := \left\{ v \in W \mid v\big|_{\partial\Omega} = u_1,\ \|\nabla v(\xi)\| \leq j_c\big(|u(\xi)|\big) \text{ a.e. in } \Omega \right\} \tag{10}$$

for a given function u_1.

In our realizations of this problem, we always take $\Omega = (0,1) \times (0,1)$ and $j_c(t) := t$. To discretize the problem, we choose a number $N \in \mathbb{N}$, a stepsize $h := \frac{1}{N+1}$, and the discrete points

$$\xi_i^{(1)} := ih = \frac{i}{N+1}, \quad \xi_j^{(2)} := jh = \frac{j}{N+1} \quad \forall\, i,j = 0,1,\ldots,N+1.$$

Furthermore, let

$$u_{i,j} := u(\xi_i^{(1)}, \xi_j^{(2)}), \quad v_{i,j} := v(\xi_i^{(1)}, \xi_j^{(2)}) \quad \forall\, i,j = 0,1,\ldots,N+1$$

and note that the values of $u_{i,j}, v_{i,j}$ are known for $i,j \in \{0, N+1\}$ due to the given boundary condition. Therefore, the discrete unknowns are the components $u_{i,j}$ for $i,j \in \{1,\ldots,N\}$. We approximate the partial gradients of u and v by using forward finite differences. Moreover, an integral of the form $\int_\Omega f(\xi)d\xi$ for a suitable function f is approximated by a piecewise constant function in such a way that we get

$$\int_\Omega f(\xi)d\xi \approx h^2 \sum_{i,j=0}^{N} f_{i,j},$$

where $f_{i,j} := f\big(\xi_i^{(1)}, \xi_j^{(2)}\big)$. We then reorder the unknows $u_{i,j}$ and define

$$\mathrm{vec}(u) := (u_{1,1}, u_{2,1}, \ldots, u_{N,1}, u_{1,2}, u_{2,2}, \ldots, u_{N,2}, \ldots, u_{N,N})^T \in \mathbb{R}^n, \quad n := N^2.$$

In a similar way, we also define $\mathrm{vec}(v)$. To get back to our standard notation, we finally set

$$x := \mathrm{vec}(u) \quad \text{and} \quad y := \mathrm{vec}(v).$$

Altogether, this results in a QVI with a linear function F of the form

$$F(x) := A^T(Ax + a) + C^T(Cx + c) \tag{11}$$

for certain matrices $A, C \in \mathbb{R}^{(n+N)\times n}$, and vectors $a, c \in \mathbb{R}^{n+N}$ (these vectors take into account the boundary conditions). Furthermore, the constraints take the form

$$g_k^P(y,x) := (A \cdot y + a)^2_{k+1+\lfloor (k-1)/N \rfloor} + (C \cdot y + c)^2_{N+k} - h^2 x_k^2 \tag{12}$$

for all $k = 1, \ldots n$, where $\lfloor \cdot \rfloor$ denotes the floor-function. The precise data of A, C, a, c are given in the corresponding Matlab files. Different instances of the discretized problems arise from different choices of the discretization parameter $N \in \mathbb{N}$ and the boundary function u_1.

KunR11 - KunR21 - KunR31 [LAO-D-2500-0-0-2500-0]

KunR12 - KunR22 - KunR32 [LAO-D-4900-0-0-4900-0]

source: [8]

description: These problems arise from the general description with F and g^P described in (11) and (12). We took $N = 50$ for problems KunR11, KunR21 and KunR31, and $N = 70$ for KunR12, KunR22 and KunR32. The boundary function is $u_1(\xi^{(1)}, \xi^{(2)}) := 1 + \xi^{(1)} + \xi^{(2)}$ for problems KunR11 and KunR12, $u_1(\xi^{(1)}, \xi^{(2)}) := 1 - \frac{\sin(2\pi\xi^{(1)})+\cos(2\pi\xi^{(2)})}{10}$ for problems KunR21 and KunR22, and $u_1(\xi^{(1)}, \xi^{(2)}) := e^{\xi^{(1)}+\xi^{(2)}}$ for problems KunR31 and KunR32. Matrices A, C and vectors a, c can be found in the corresponding Matlab source files.

JF: positive definite (everywhere)

References

[1] M.B. Donato, M. Milasi, and C. Vitanza: *An existence result of a quasi-variational inequality associated to an equilibrium problem.* Journal of Global Optimization 40, 2008, pp. 87–97.

[2] F. Facchinei and C. Kanzow: *Generalized Nash equilibrium problems.* Annals of Operations Research 175, 2010, pp. 177–211.

[3] F. Facchinei and C. Kanzow: *Penalty methods for the solution of generalized Nash equilibrium problems.* SIAM Journal on Optimization 20, 2010, pp. 2228–2253.

[4] F. Facchinei, C. Kanzow, and S. Sagratella: *Solving quasi-variational inequalities via their KKT-conditions.* Preprint 305, Institute of Mathematics, University of Würzburg, Würzburg, Germany, January 2012.

[5] F. Facchinei and J.-S. Pang: *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volumes I and II.* Springer, 2003.

[6] M. Fukushima: *A class of gap functions for quasi-variational inequality problems.* Journal of Industrial and Management Optimization 3, 2007, pp. 165–171.

[7] M. Hintermüller and C.N. Rautenberg: *A sequential minimization technique for elliptic quasi-variational inequalities with gradient constraints.* IFP-Report No. 45, Institute of Mathematics and Scientific Computing, University of Graz, Austria, June 2011.

[8] M. Kunze and J.F. Rodrigues: *An elliptic quasi-variational inequality with gradient constraints and some of its applications.* Mathematical Methods in the Applied Sciences 23, 2000, pp. 897–908.

[9] J.P. Luna, C. Sagastizábal, and M.V. Solodov: *A class od Dantzig-Wolfe type decomposition methods for variational inequality problems.* IMPA Preprint A706, Instituto de Matematica Pura e Aplicada, Rio de Janeiro, November 2011.

[10] Y. Nesterov and L. Scrimali: *Solving strongly monotone variational and quasi-variational inequalities.* CORE Discussion Paper 2006/107, Catholic University of Louvain, Center for Operations Research and Econometrics, December 2006.

[11] M. Noor and K. Noor: *On general quasi-variational inequalities.* Journal of King Saud University, 2011, in press.

[12] J.V. Outrata, M. Kočvara, and J. Zowe: *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints.* Kluwer Academic Publishers, 1998.

[13] J.V. Outrata and J. Zowe: *A Newton method for a class of quasi-variational inequalities.* Computational Optimization and Applications 4, 1995, pp. 5–21.

[14] J.-S. Pang and M. Fukushima: *Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games.* Computational Management Science 2, 2005, pp. 21–56 (Erratum: ibid 6, 2009, pp. 373–375)

[15] I.P. Ryazantseva: *First-order methods for certain quasi-variational inequalities in Hilbert space.* Computational Mathematics and Mathematical Physics 47, 2007, pp. 183–190.

[16] L. Scrimali: *Quasi-variational inequality formulation of the mixed equilibrium in mixed multiclass routing games.* CORE Discussion Paper 2007/7, Catholic University of Louvain, Center for Operations Research and Econometrics, January 2007.