A Dantzig-Wolfe Decomposition Method for Quasi-Variational Inequalities

Manoel Jardim[1*], Claudia Sagastizábal[2] and Mikhail Solodov[1]

[1]IMPA, Estrada Dona Castorina, Rio de Janeiro, 22460-320, RJ, Brasil.
[2]IMECC, Unicamp, Rua Sérgio Buarque de Holanda, Campinas, 13083-859, SP, Brasil.

*Corresponding author(s). E-mail(s): manoel.jardim@impa.br;
Contributing authors: sagastiz@unicamp.br; solodov@impa.br;

Abstract

We propose an algorithm to solve quasi-variational inequality problems, based on the Dantzig-Wolfe decomposition paradigm. Our approach solves in the subproblems variational inequalities, which is a simpler problem, while restricting quasi-variational inequalities in the master subproblems, making them generally (much) smaller in size when the original problem is large-scale. We prove global convergence of our algorithm, assuming that the mapping of the quasi-variational inequality is either single-valued and continuous or it is set-valued maximally monotone. Quasi-variational inequalities serve as a framework for several equilibrium problems, and we apply our algorithm to an important example in the field of economics, namely the Walrasian equilibrium problem formulated as a generalized Nash equilibrium problem. Our numerical assessment demonstrates good performance and usefullness of the approach for the large-scale cases.

Keywords: Quasi-variational inequality; Dantzig-Wolfe decomposition; Variational inequality; Walrasian equilibrium problem; Generalized Nash equilibrium problem.

1 Introduction

The framework provided by a quasi-variational inequality (QVI) setting encompasses many problems related to optimization and equilibrium; see [1]. The goal of a QVI problem is to find a pair (x^*, z^*) such that

$$x^* \in K(x^*) \text{ with } z^* \in F(x^*) \text{ satisfy } \langle z^*, y - x^* \rangle \geq 0 \text{ for all } y \in K(x^*). \tag{1}$$

In these relations, $\langle \cdot, \cdot \rangle$ stands for the Euclidean inner product in $\mathbb{R}^n$ and $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ and $K : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ are two maps associating vectors in $\mathbb{R}^n$ to subsets of $\mathbb{R}^n$. A variational inequality (VI) corresponds to a less general case, with K in (1) being a constant set-valued map, i.e., $K(x) = D \subset \mathbb{R}^n$ for all $x \in \mathbb{R}^n$. In most meaningful VI settings, in particular those subsuming local optimality conditions in optimization, the set D is convex. If D is further the nonnegative orthant, the corresponding VI is a nonlinear complementarity problem. The so-called mixed complementarity problem arises when the set D is a generalized box (given component-wise by one-sided or two-sided bound constraints, or no constraints at all for some components). For details and related discussions, we refer again to [1].

The introduction in [2] provides a good review of the history of QVIs. We recall here that the first work on the subject dates back to [3], where impulsive control problems were formulated according to the format (1). The framework has proven effective in handling important applications in engineering [4, 5], transportation [6], and economics [7]. In particular, QVIs offer a favorable environment for modeling equilibria, of the Radner type as in [8], or resulting from a generalized Nash game [9].

Much of QVI literature focuses mainly on theoretical results, especially concerning the existence of solutions. Algorithmic research on the subject is less developed, probably due to certain inherent difficulties associated to QVIs. The proposal in [10], to solve (1) by minimizing a nonsmooth gap function, does not present a specific solution algorithm. Inspired by the structure of generalized Nash games, [10] solves sequentially VIs that are shown to converge to a solution of (1). The scheme was revisited and enhanced in [11–13]. Newtonian approaches for solving the system of optimality conditions derived from (1) were considered in the works [14, 15], reporting only some limited numerical experiments. The specialized potential reduction interior point method proposed in [16] exhibits good performance on the test library in [17].

It is, nonetheless, unclear how well the aforementioned algorithms scale for large instances. In mathematical optimization, decomposition methods are best suited to handle large problems. To provide a first step in this direction when dealing with QVIs, we introduce a Dantzig-Wolfe-like method that is shown to converge to a solution of (1) under natural assumptions. Borrowing the linear programming terminology, the proposed algorithm alternates between solving a simple QVI as its "master program" and a VI as its "subproblem". Thus, the approach departs from the sequential solution of VIs that has been predominant in the literature. The motivation is to tailor the method to facilitate subproblem separability, crucial when dimensions in (1) are large.

We mention, in passing, that sometimes QVIs can be reduced to a VI, as is the case of variational equilibria of some Nash games [18, 19]. For a discussion of related issues

in the context of energy markets we refer to [20]. Furthermore, the reproducible set-valued maps introduced in [21] identify situations in which solving a VI provides all the solutions of a QVI. But these are very specific configurations, far from representing general situations. Indeed, QVIs have a significantly more complex structure than VIs, and therefore more sophisticated tools and developments are needed to solve problems such as (1) using decomposition techniques.

The Dantzig-Wolfe (DW) decomposition method was introduced in [22] to solve large linear programs having a structured feasible set, whose constraints can be separated into "hard" and "easy" ones (generally separable). Each iteration solves a master program followed by a subproblem, respectively aiming at guaranteeing feasibility and objective function decrease. The master program outputs multipliers associated with the hard constraints, which define the subproblem objective function. The subproblem then has an "easy" feasible set (usually yielding separable subproblems). The subproblem solution is informed to the master program, so that in the next iteration the approximation of the master's feasible set is improved. Because subproblems can often be solved in parallel, the computational burden is dominated by the master program solution, whose size grows at each iteration.

The DW decomposition for VIs was introduced in [23, 24]. The approach was extended and improved in [25], including several theoretical generalizations and an application to large-scale generalized Nash games. Another use of DW techniques of [25] is given in [26], where a class of risk-averse stochastic equilibrium problems is considered, numerically assessed on the "real-world" European market of natural gas. Benders decomposition for VIs, which is the dual approach to DW, is presented in [27].

In this work, we show how the DW approach can be applied to very general QVIs. When iterating between solving QVI-master problems and VI-subproblems sequentially, our DW method computes in the process a certain gap function that provides information about convergence. As illustrations, we start by computing the Walrasian equilibrium of large economies, a very classical and important problem in economics, recently considered in [28, 29]. We afterwards solve also some academic QVI examples from [16, 17], said to have "moving sets". In both cases, our approach gives the same results as the commercial solver GAMS [30], with substantially less computational effort for the larger dimensions. Over many random instances generated for the Walrasian equilibrium problem, the DW method computing times are not only shorter but consistently less volatile than those for the direct solution by GAMS.

The rest of the paper is organized as follows. In Section 2, we give the mathematical formulation of our DW algorithm. Section 3 is devoted to convergence results. When the problem has a certain special structure, Section 4 gives details about techniques to uncouple variables and make subproblems separable. Presenting the two sets of numerical experiments (Walrasian equilibrium and moving set example), the results in Section 5 provide empirical evidence of the good performance of our method on large-scale problems, when compared with the direct application of GAMS software.

Some final words about our notation and terminology. The Euclidean inner product in $\mathbb{R}^n$ is denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ is the associated norm. By $\| \cdot \|_\infty$ we denote the maximum norm. For a set $D \subset \mathbb{R}^n$, its interior is denoted by $\operatorname{int}(D)$ and its convex hull (the smallest convex set in $\mathbb{R}^n$ that contains D), is denoted by $\operatorname{Co} D$. For a convex

set D, the notation $\mathcal{N}_D(x) = \{u : \langle u, y - x\rangle \leq 0,\ \forall\, y \in D\}$ stands for the normal cone to D at x when $x \in D$ ($\mathcal{N}_D(x) = \emptyset$ otherwise). The mapping $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is strongly monotone if there exists $c > 0$ such that $\langle u - v, x - y\rangle \geq c\|x - y\|^2$ for all $x, y \in \mathrm{dom}(F) = \{z \in \mathbb{R}^n : F(z) \neq \emptyset\}$ and all $u \in F(x)$, $v \in F(y)$. Also, F is monotone if the above inequality holds for $c = 0$. A monotone set-valued mapping $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is maximally monotone if its graph $\{(x, u) \in \mathbb{R}^n \times \mathbb{R}^n : x \in \mathrm{dom}(F),\ u \in F(x)\}$ is not properly contained in the graph of any other monotone mapping. We also use the properties that a maximally monotone operator F is both locally bounded in $\mathrm{int}(\mathrm{dom}(F))$ [31], and outer-semicontinuous [32, Chapter 4].

2 Dantzig-Wolfe decomposition for QVI

Let there be given functions $h : \mathbb{R}^n \to \mathbb{R}^l$ and $g(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$, with h convex, and $g(\cdot, x)$ convex and differentiable for each $x \in \mathbb{R}^n$, and the set in (1) be given by

$$K(x) = K_g(x) \cap K_h, \text{ where } \begin{array}{ll} K_g(x) & = \{y \in \mathbb{R}^n : g(y, x) \leq 0\} \\ K_h & = \{y \in \mathbb{R}^n : h(y) \leq 0\}\ . \end{array} \tag{2}$$

The QVI setting is a good candidate for the DW technique, as in (2) "difficult" constraints can be considered those involving both variables x and y, while the constraints in y are naturally "easy" (or at least easier).

We assume that $K_h \subset \mathrm{int}(\mathrm{dom}(F))$, and that some $y^1 \in K_g(y^1) \cap K_h$ is given, to start the process.

2.1 General organization and master QVI definition

In (2), it seems easier to ensure feasibility with respect to K_h. Since the x-parameterized constraints $g(y, x)$ are naturally harder to handle, they are dealt with in the master. Similarly to the DW scheme in linear programming, the corresponding multiplier associated to the parametrized constraint provides its Lagrangian relaxation, which is incorporated into the definition of the operator in the subproblem, whose feasible set is K_h, and it is therefore a VI.

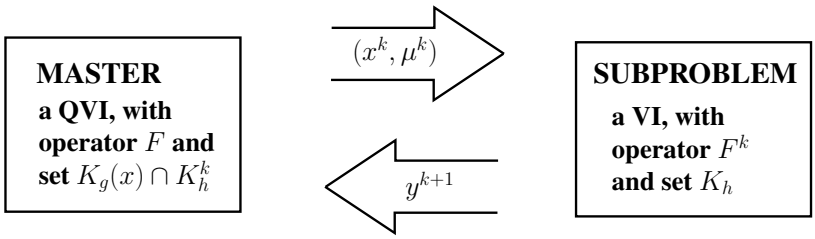

Fig. 1 Some elements of the DW decomposition for (1)-(2). On the left, the master problem solves a QVI that outputs x^k and a multiplier μ^k associated to the g-constraints. This primal-dual pair is used by the subproblem on the right to define the operator F^k, and return a solution y^{k+1} to the master problem. The output of the subproblem is used by the master in the next iteration, to define the set K_h^{k+1}.

To introduce gradually the notation, the scheme in Figure 1 reports the information exchanged between the master problem and the subproblem at each iteration k of the process. The information generated when solving the VI subproblem at iteration k, which is y^{k+1}, is used by the master QVI at iteration $k+1$ to approximate the set K_h. The approximation takes the convex hull of the output from the past VI-subproblems.

When formulating the master QVI solved at iteration k, the last subproblem information available to the master is y^k. Accordingly, the QVI master (3) at iteration k finds a pair (x^k, z_m^k) for which

$$\begin{array}{rlll} x^k \in K_g(x^k) \cap K_h^k,\ z_m^k \in F(x^k) & \text{satisfy} & \langle z_m^k, x - x^k \rangle \geq 0 \\ & \text{for all} & x \in K_g(x^k) \cap K_h^k \\ \text{where } K_h^k = \operatorname{Co} Y^k & \text{and} & Y^k = \{y^1, \ldots, y^k\}\ . \end{array} \tag{3}$$

Another part of the output of the QVI master (3) is a multiplier μ^k associated to the g-constraints. Because the intersection with $\operatorname{Co} Y^k$ reduces the search for feasible elements to a $(k-1)$-dimensional simplex, the master problem (3) is a relatively simple QVI even if n is very large.

2.2 Subproblem definition

As shown by Figure 1, after the master (3) is solved, the VI-subproblem receives in addition to x^k, information on μ^k, the multiplier associated with the constraints defining $K_g(x^k)$. The pair (x^k, μ^k) is used to define the subproblem operator F^k, for example as

$$F^k(y) = F(y) + \sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, x^k)\,. \tag{4}$$

Other choices are possible, depending on how the first and second terms, respectively involving F and $\nabla_y g$, are approximated. For the latter, notice that (4) makes a constant approximation of the gradient of the hard constraint. The approximation could also let the first variable free or, to have a more flexible framework, both options could be combined using a parameter $\omega_i^k \in [0,1]$. It is then notationally convenient to write the second term in (4) in matrix-vector form. Thus, we introduce a (transposed Jacobian) matrix $\Gamma^k(y)$ of order $n \times m$ with columns defined as below, for $j = 1, \ldots, m$:

$$\Gamma_j^k(y) = \begin{cases} \nabla_y g_j(x^k, x^k) & \text{(constant)} \\ \nabla_y g_j(y, x^k) & \text{(free)} \\ \omega_j^k\ \nabla_y g_j(x^k, x^k) + (1 - \omega_j^k) \nabla_y g_j(y, x^k) & \text{(convex combination).} \end{cases} \tag{5}$$

Some comments regarding (5) are in order. When the product $\Gamma^k(y)\mu^k$ is computed using the (constant) first option above, it coincides with the expression for the second term in (4). The second option in (5) can be advantageous when we aim to retain more information about the g constraints in the subproblem formulation. For our instances of Walrasian equilibrium in Section 5, the function g depends linearly on y (see (33)), and then all the options become the same. In general though, this need not be the case, and having this feature adds to the broader flexibility and applicability of the

framework. The impact of the third option is assesed in the numerical experiments on moving set problems in Subsection 5.2. Finally, notice that, although not made explicit in the notation, the third option involves the parameter $\omega^k \in [0,1]^m$.

A similar approach can be employed for the first term in (4), that is, approximating F by an operator $\widehat{F}^k$ that uses information output by the master QVI. Along the lines in [25], the approximation can be constant, of first-order, or the actual mapping:

$$\widehat{F}^k(y) = \begin{cases} \{z_m^k\} & \text{for } z_m^k \in F(x^k) \text{ from (3)} \quad \text{(constant)} \\ F(x^k) + \nabla F(x^k)(y - x^k), & \text{if } F \in C^1 \text{ is single-valued} \quad \text{(first order)} \\ F(y) & \quad \text{(exact).} \end{cases} \tag{6}$$

Finally, if needed or advantageous, the VI-subproblem operator can also be regularized using an $n \times n$ positive (semi)definite matrix Q^k.

In full generality, the VI subproblem (7) at iteration k finds a pair (y^{k+1}, z_s^{k+1}) for which

$$\begin{array}{rl} y^{k+1} \in K_h\,,\ z_s^{k+1} \in F^k(y^{k+1}) \text{ satisfy } & \left\langle z_s^{k+1}, y - y^{k+1} \right\rangle \geq 0 \ \text{ for all } y \in K_h \\ \text{where } & F^k(y) = \widehat{F}^k(y) + \Gamma^k(y)\mu^k + Q^k(y - x^k) \\ & \text{with } \widehat{F}^k \text{ from (6) and } \Gamma^k \text{ from (5).} \end{array} \tag{7}$$

For later use notice that, after solving (3) and (7), the inclusions

$$\begin{array}{rcl} \zeta^k & = & z_m^k + \Gamma^k(x^k)\mu^k \in F^k(x^k) \text{ and} \\ \widehat{\zeta}^k & = & z_s^{k+1} - \Gamma^k(y^{k+1})\mu^k - Q^k(y^{k+1} - x^k) \in \widehat{F}^k(y^{k+1}) \end{array} \tag{8}$$

hold for all the approximating variants proposed in (5) and (6).

Both QVI-master (3) and VI-subproblem (7) are assumed to be solvable at all iterations. We shall not go into extensive discussions on sufficient conditions for that assumption to hold, because they are well known. For the VI subproblems, in particular, solvability follows if the set K_h is compact. Regardless of compactness of that set, subproblems are always solvable if the approximation F^k in (7) is strongly monotone, a property that can be ensured in our setting, as explained next.

Remark 1 (On strong monotonicity of subproblem operators)**.** The approximating mapping F^k in (7) can always be chosen to be strongly monotone.

Note that F^k involves three terms, the first one being the approximation $\widehat{F}^k$ defined in (6). If F is monotone, any option in (6) preserves monotonicity. If F is not monotone, the constant option in (6) makes the term $\widehat{F}^k$ monotone. Regarding the second term defining F^k, it is always monotone because, by (5) and by the convexity of $g_i(\cdot, x^k)$, it holds that

$$\left\langle \Gamma^k(y)\mu^k - \Gamma^k(x)\mu^k, y - x \right\rangle = \sum_{i=1}^{m} \mu_i^k (1 - \omega_i^k) \left\langle \nabla_y^k g_i(y, x^k) - \nabla_y g_i(x, x^k), y - x \right\rangle \geq 0.$$

Since the first two terms in F^k are monotone, and in the third term we can take a positive definite matrix Q^k (if needed), the claim of strong monotonicity of F^k follows. □

3 Algorithm statement and its convergence

After the master QVI is solved, the information x^k, z_m^k, μ^k is available. For $y \in \mathbb{R}^n$, we define the gap function

$$\text{GAP}(y) = \left\langle \zeta^k, y - x^k \right\rangle \text{ with } \zeta^k = z_m^k + \sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, x^k) \in F^k(x^k), \tag{9}$$

where the inclusion $\zeta^k \in F^k(x^k)$ is by (8), recalling also (5) to evaluate ζ^k.

Convergence of our DW decomposition method is determined as in the linear programming setting, by monitoring the value of the gap function at y^{k+1}, the subproblem solution. The framework of the method is outlined in Algorithm 1.

Algorithm 1 Dantzig-Wolfe decomposition for QVI

Require: $y^1 \in K_g(y^1) \cap K_h$.
Ensure: Accumulation points of the iterates solving QVI (1)-(2).
  Set $\text{GAP}^1 = -\infty$ and $k \leftarrow 1$.
  while $\text{GAP}^k < 0$ **do**
    MASTER SOLUTION: solve (3) to compute the pair (x^k, z_m^k) and the multiplier μ^k.
    SUBPROBLEM SOLUTION: solve (7) to compute y^{k+1}.
    STOPPING CRITERION: Compute $\text{GAP}^{k+1} = \text{GAP}(y^{k+1})$ defined in (9).
    UPDATE: $k \leftarrow k + 1$
  end while

In Theorem 4 we shall prove that accumulation points of the iterates generated by Algorithm 1 solve QVI (1)-(2).

We first state the Karush-Kuhn-Tucker (KKT) conditions for (3), which among other things specify the multiplier μ^k employed in both (7) and (9).

Theorem 1 (KKT conditions for master QVI)**.** *Under any suitable constraint qualification, if the pair* $\left(x^k, z_m^k \in F(x^k)\right)$ *solves the master QVI* (3) *at iteration* k*, then there exists a Lagrange multiplier* $\mu^k \in \mathbb{R}^m$ *such that*

$$0 \in z_m^k + \sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, x^k) + \mathcal{N}_{K_h^k}(x^k), \tag{10}$$

$$\mu_i^k \geq 0, \quad g_i(x^k, x^k) \leq 0, \quad \mu_i^k g_i(x^k, x^k) = 0, \; i = 1, \ldots, m. \quad \square \tag{11}$$

We do not include a proof of this result, referring to the corresponding theorem in [2] in a more general setting of QVIs in Banach spaces, under the Robinson's constraint qualification. For other suitable constraint qualifications in finite dimensions, see [33].

3.1 Gap function and finite termination

Convergence of Algorithm 1 relies on the stopping criterion GAP^k being driven to zero by the iterative process. For this reason, we start by examining the properties of

the gap function. Recall that according to Remark 1, the mapping F^k can always be chosen to be strongly monotone.

Lemma 2 (Gap properties)**.** *Let F^k be strongly monotone. At each iteration k of Algorithm 1, the following holds for the gap function defined in (9):*

1. $\text{GAP}(y^{k+1}) \leq 0$*;*
2. $\text{GAP}(y) \geq 0$ *for all* $y \in K_h^k$*;*
3. $\text{GAP}(y^{k+1}) = 0$ *if and only if* $x^k = y^{k+1}$*; and*
4. *if* $\text{GAP}(y^{k+1}) < 0$ *then* $y^{k+1} \notin K_h^k$*. Hence,* $K_h^k \subsetneq K_h^{k+1}$*.*

Proof. Let $\left(y^{k+1}, z_s^{k+1} \in F^k(y^{k+1})\right)$ be the pair solving the VI subproblem. The first inclusion from (8) states that $\zeta^k = z_m^k + \Gamma^k(x^k)\mu^k \in F^k(x^k)$, so

$$z_s^{k+1} \in F^k(y^{k+1}) \text{ and } \zeta^k \in F^k(x^k) \Longrightarrow \left\langle z_s^{k+1} - \zeta^k, y^{k+1} - x^k \right\rangle \geq 0\,, \tag{12}$$

by the monotonicity of F^k. Recalling the gap definition (9), we obtain that

$$\text{GAP}(y^{k+1}) = \left\langle \zeta^k, y^{k+1} - x^k \right\rangle \leq \left\langle z_s^{k+1}, y^{k+1} - x^k \right\rangle .$$

Because $x^k \in K_h$, the inequality in the first line in (7) yields $\left\langle z_s^{k+1}, x^k - y^{k+1} \right\rangle \geq 0$. We conlude that

$$\text{GAP}(y^{k+1}) \leq \left\langle z_s^{k+1}, y^{k+1} - x^k \right\rangle \leq 0, \tag{13}$$

proving item 1.

To show item 2, let $\nu^k \in \mathcal{N}_{K_h^k}(x^k)$ be the normal element in Theorem 1 that makes the inclusion (10) an equality:

$$0 = z_m^k + \sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, x^k) + \nu^k = \zeta^k + \nu^k, \text{ with } \left\langle \nu^k, y - x^k \right\rangle \leq 0 \text{ for all } y \in K_h^k\,.$$

By definition (9), we have that $\text{GAP}(y) = -\left\langle \nu^k, y - x^k \right\rangle$ for all $y \in \mathbb{R}^n$. In particular,

$$\text{GAP}(y) = -\left\langle \nu^k, y - x^k \right\rangle \geq 0 \text{ for all } y \in K_h^k\,,$$

which proves item 2.

To continue with item 3, suppose that $x^k = y^{k+1}$. Then $\text{GAP}(y^{k+1}) = \left\langle \zeta^k, y^{k+1} - x^k \right\rangle = 0$, recalling once again the gap definition in (9). Conversely, when $\text{GAP}(y^{k+1}) = \left\langle \zeta^k, y^{k+1} - x^k \right\rangle = 0$, we see from (13) that $\left\langle z_s^{k+1}, y^{k+1} - x^k \right\rangle = 0$. In (12) when have that $\left\langle z_s^{k+1} - \zeta^k, y^{k+1} - x^k \right\rangle = 0$, which implies that $x^k = y^{k+1}$, by strong monotonicity of F^k.

For the final item, for the sake of contradiction, assume that $y^{k+1} \in K_h^k$. Then, by item 2, $\text{GAP}(y^{k+1}) \geq 0$. This contradicts the hypothesis $\text{GAP}(y^{k+1}) < 0$, concluding the proof of item 4. □

We can now show convergence when the method terminates finitely. Note that according to Lemma 2 we have that $\text{GAP}^k \leq 0$ for all k, and therefore, Algorithm 1 stopping finitely at some iteration k means that $\text{GAP}^{k+1} = 0$ is computed.

Corollary 3 (Finite termination)**.** *If at some iteration k of Algorithm 1, $\text{GAP}^{k+1} = 0$ is computed, then the algorithm stops. In this case, the pair $\left(x^k, \widehat{\zeta}^k\right)$, with x^k solving the master problem (3) and $\widehat{\zeta}^k$ from (8), solves the original QVI, given by (1) and (2).*

Proof. By item 3 in Lemma 2, having $\text{GAP}(y^{k+1}) = 0$ is equivalent to $x^k = y^{k+1}$. In particular,

$x^k \in K_g(x^k) \cap K_h$ because x^k solves (3) and (7), and
$\widehat{\zeta}^k \in F(x^k)$ because $\widehat{\zeta}^k \in \widehat{F}^k(y^{k+1}) = \widehat{F}^k(x^k) \subset F(x^k)$ for any choice in (6).

To verify that the pair $\left(x^k, \widehat{\zeta}^k \in F(x^k)\right)$ solves (1)-(2), it only remains to show that

$$\left\langle \widehat{\zeta}^k, y - x^k \right\rangle \geq 0 \text{ for all } y \in K_g(x^k) \cap K_h \,. \tag{14}$$

Since x^k solves the VI subproblem (7), it holds that

$$\left\langle z_s^{k+1}, y - x^k \right\rangle \geq 0 \;\text{ for all } y \in K_h, \text{ in particular for all } y \in K_g(x^k) \cap K_h \,.$$

Then, as $z_s^{k+1} = \widehat{\zeta}^k + \Gamma^k(x^k)\mu^k$ by the definition of $\widehat{\zeta}^k$ in (8), it holds that

$$\left\langle \widehat{\zeta}^k + \Gamma^k(x^k)\mu^k, y - x^k \right\rangle \geq 0 \;\text{ for all } y \in K_g(x^k) \cap K_h \,.$$

The inequality (14) will hold if

$$\left\langle \Gamma^k(x^k)\mu^k, y - x^k \right\rangle \leq 0 \;\text{ for all } y \in K_g(x^k) \cap K_h \,. \tag{15}$$

First recall that $\Gamma^k(x^k)\mu^k = \sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, x^k)$. Second, note that because the functions $g_i(\cdot, x^k)$ are convex, for all $y \in \mathbb{R}^n$ we have that

$$\left\langle \nabla_y g_i(x^k, x^k), y - x^k \right\rangle \leq g_i(y, x^k) - g_i(x^k, x^k) \,.$$

Then multiplying this inequality by $\mu_i^k \geq 0$ and using the condition $\mu_i^k g_i(x^k, x^k) = 0$ from (11) in Theorem 1 yields, for any $y \in K_g(x^k) \cap K_h$, that

$$\mu_i^k \left\langle \nabla_y g_i(x^k, x^k), y - x^k \right\rangle \leq \mu_i^k g_i(y, x^k) - \mu_i^k g_i(x^k, x^k) = \mu_i^k g_i(y, x^k) \,.$$

Since $y \in K_g(x^k)$, the right-hand side in the relation above is nonpositive. Summing up these inequalities for $i = 1, \ldots, m$ yields (15), as claimed. □

3.2 Asymptotic convergence of Algorithm 1

Corollary 3 states that whenever the master QVI solution is also a solution to the VI subproblem, the algorithm stops having found a solution to the original QVI. Otherwise, the iterative process continues and, by item 4 in Lemma 2, the algorithm

makes progress by defining a larger feasible set for the master QVI (3), better approximating the original problem. To complete our convergence analysis, it remains to consider the asymptotic behavior of Algorithm 1. Since now we are dealing with an infinite number of iterations, the strong monotonicity of the approximations discussed in Remark 1 needs to be ensured in a uniform manner, for all iterates. Before stating the result, we note that existence results about QVI solutions usually require some kind of monotonicity or continuity of F, assumptions also used below.

Theorem 4. *Suppose that the operator F in (1) is either continuous and single-valued or maximally monotone set-valued, and that the function g in (2) has a continuous gradient $\nabla_y g(\cdot,\cdot)$. Let the VI-operator approximations F^k in (7) be defined to be uniformly strongly monotone with parameter $c > 0$, and let the matrices $\{Q^k\}$ be taken bounded. Finally, suppose that if Algorithm 1 generates an infinite number of itarations, the sequences $\{\mu^k\}$ and $\{y^{k+1}\}$ are bounded.*

Then,

$$\lim_{k\to\infty} \textsc{gap}(y^{k+1}) = 0, \quad \lim_{k\to\infty} \|y^{k+1} - x^k\| = 0,$$

and every accumulation point of $\left\{\left(x^k, \widehat{\zeta}^k\right)\right\}$ is a solution to QVI (1)-(2).

Proof. By Lemma 2 and Corollary 3, for Algorithm 1 to make an infinite number of iterations, it must hold that $\textsc{gap}(y^{k+1}) < 0$ for all k. Suppose the claim were not true, i.e., $\liminf_{k\to\infty} \textsc{gap}(y^{k+1}) < 0$. Then there exist $\varepsilon > 0$ and an infinite subset of iterations $\mathbb{N}_\varepsilon$, such that $\textsc{gap}(y^{k+1}) \le -\varepsilon$ for all $k \in \mathbb{N}_\varepsilon$. Recalling (9), for this subsequence it holds that

$$\left\langle z_m^k + \sum_{i=1}^m \mu_i^k \nabla_y g_i(x^k, x^k), y^{k+1} - x^k \right\rangle \le -\varepsilon\,. \tag{16}$$

Consider $k, j \in \mathbb{N}_\varepsilon$, with $j > k$. As x^j solves the QVI master (3) at iteration j, by (10) in Theorem 1 there exists $\nu^j \in \mathcal{N}_{\mathrm{Co}\, Y^j}(x^j)$ such that

$$0 = z_m^j + \sum_{i=1}^m \mu_i^j \nabla_y g_i(x^j, x^j) + \nu^j. \tag{17}$$

By the construction of (3), and since $j > k$, we have that $y^{k+1} \in \mathrm{Co}\, Y^j$, and hence,

$$\left\langle \nu^j, y^{k+1} - x^j \right\rangle \le 0.$$

Combining the latter relation with (17), we obtain that

$$\left\langle z_m^j + \sum_{i=1}^m \mu_i^j \nabla_y g_i(x^j, x^j), y^{k+1} - x^j \right\rangle \ge 0. \tag{18}$$

As $\{y^k\}$ is bounded, the sets Y^k are uniformly bounded in k, and hence so are the sets $K_g(\cdot) \cap K_h^k$ in (3). It follows that the sequence $\{x^k\}$ is bounded. Let $\bar{x}$ be any

accumulation point of the (bounded) subsequence $\{x^j\}$, $j \in \mathbb{N}_\varepsilon$. Without ambiguity for further developments, we can assume that the whole $\{x^j\}$, $j \in \mathbb{N}_\varepsilon$, converges to some $\bar{x}$ (otherwise, just pass onto a subsequence within $\mathbb{N}_\varepsilon$, and re-define $\mathbb{N}_\varepsilon$). We can then assume that $\{\mu^j\}$, $j \in \mathbb{N}_\varepsilon$, converges to some $\bar{\mu}$ (again, passing onto a further subsequence, if necessary).

If F is continuous single-valued, then $z_m^j = F(x^j) \to F(\bar{x}) = \bar{z}_m$, $j \in \mathbb{N}_\varepsilon$. If F is a maximally monotone set-valued mapping, then it is locally bounded in K_h because $K_h \subset \operatorname{int}(\operatorname{dom}(F))$, and it is also outer-semicontinuous. Then, again passing onto a further subsequence if necessary, we can assume that $z_m^j \to \bar{z}_m \in F(\bar{x}), j \in \mathbb{N}_\varepsilon$.

Then, taking the limit $j \to \infty, j \in \mathbb{N}_\varepsilon$, in (18), we conclude that

$$\left\langle \bar{z}_m + \sum_{i=1}^{m} \bar{\mu}_i \nabla_y g_i(\bar{x}, \bar{x}), y^{k+1} - \bar{x} \right\rangle \geq 0, \quad \bar{z}_m \in F(\bar{x}). \tag{19}$$

Again, passing onto a subsequence if necessary, we can assume that $\{y^{k+1}\}$, $k \in \mathbb{N}_\varepsilon$, converges to some $\bar{y}$. Taking this limit in (16), we obtain that

$$\left\langle \bar{z}_m + \sum_{i=1}^{m} \bar{\mu}_i \nabla_y g_i(\bar{x}, \bar{x}), \bar{y} - \bar{x} \right\rangle \leq -\varepsilon,$$

while taking the same limit in (19) yields

$$\left\langle \bar{z}_m + \sum_{i=1}^{m} \bar{\mu}_i \nabla_y g_i(\bar{x}, \bar{x}), \bar{y} - \bar{x} \right\rangle \geq 0.$$

This contradiction shows that $\liminf_{k\to\infty} \textsc{gap}(y^{k+1}) \geq 0$, which means that $\lim_{k\to\infty} \textsc{gap}(y^{k+1}) = 0$, as $\textsc{gap}(y^{k+1}) < 0$ for all k.

Since y^{k+1} solves the VI subproblem (7) at iteration k and $x^k \in K_h$, it holds that

$$\left\langle z_s^{k+1}, x^k - y^{k+1} \right\rangle \geq 0, \tag{20}$$

with $z_s^{k+1} \in F^k(y^{k+1})$ by (7). As F^k is uniformly strongly monotone (with modulus c),

$$\left\langle \zeta^k - z_s^{k+1}, x^k - y^{k+1} \right\rangle \geq c \left\| x^k - y^{k+1} \right\|^2,$$

because $\zeta^k \in F^k(x^k)$ by (8). As a result, recalling (8), we have that

$$\left\langle z_m^k + \Gamma^k(x^k)\mu^k - z_s^{k+1}, x^k - y^{k+1} \right\rangle \geq c \left\| x^k - y^{k+1} \right\|^2. \tag{21}$$

Combining (21) and (20), and recalling (9), we obtain that

$$-\textsc{gap}(y^{k+1}) \geq c \left\| x^k - y^{k+1} \right\|^2. \tag{22}$$

Since $\textsc{gap}(y^{k+1}) \to 0$ as $k \to \infty$ (as established previously), it follows that $\|x^k - y^{k+1}\| \to 0$ as $k \to \infty$.

Let $(\bar{x}, \bar{\zeta})$ be an accumulation point of $\{(x^k, \widehat{\zeta}^k)\}$. Since $x^k \in K_g(x^k) \cap K_h^k \subset K(x^k)$, and g and h are continuous, we have $\bar{x} \in K(\bar{x})$. We can take a subsequence $\{k_j\}$ such that $x^{k_j} \to \bar{x}$, $\omega^{k_j} \to \bar{\omega}$, $\mu^{k_j} \to \bar{\mu}$ and $Q^{k_j} \to \bar{Q}$, as $j \to \infty$.

As

$$\|\bar{x} - y^{k_j+1}\| \leq \|\bar{x} - x^{k_j}\| + \|x^{k_j} - y^{k_j+1}\|,$$

taking the limit as $j \to \infty$, we obtain that $y^{k_j+1} \to \bar{x}$.

Let $y \in K(\bar{x})$ be arbitrary. It holds that $y \in K_h$. As $(y^{k_j+1}, z_s^{k_j+1}) \in K_h \times F^{k_j}(y^{k_j+1})$ solves the VI subproblem (7) at iteration k_j, we have that

$$\left\langle z_s^{k_j+1}, y - y^{k_j+1} \right\rangle \geq 0 \quad \text{and} \quad z_s^{k_j+1} \in F^{k_j}(y^{k_j+1}). \tag{23}$$

Recalling the relation $\widehat{\zeta}^{k_j} = z_s^{k_j+1} - \Gamma^{k_j}(y^{k_j+1})\mu^{k_j} + Q^{k_j}(y^{k_j+1} - x^{k_j})$ from (8), we obtain that

$$\langle \widehat{\zeta}^{k_j} + \Gamma^{k_j}(y^{k_j+1})\mu^{k_j} - Q^{k_j}(y^{k_j+1} - x^{k_j}), y - y^{k_j+1} \rangle \geq 0. \tag{24}$$

Note that $\Gamma^{k_j}(y^{k_j+1})\mu^{k_j}$ tends to $\sum_{i=1}^{m} \bar{\mu}_i \nabla_y g_i(\bar{x}, \bar{x})$ for any choice in (5), and $Q^{k_j}(y^{k_j+1} - x^{k_j})$ tends to zero, as $j \to \infty$. If F is a continuous single-valued map, then for any choice of $\widehat{F}^k$ in (6), $\widehat{\zeta}^{k_j} = \widehat{F}^{k_j}(y^{k_j+1})$ tends to $\bar{\zeta} = F(\bar{x})$. For the set-valued case, since F is maximally monotone, it is outer semicontinuous and locally bounded. Then any accumulation point of any elements in $\{\widehat{F}^{k_j}(y^{k_j+1})\}$ belongs to $F(\bar{x})$. In either case, passing onto a subsequence in the second case if necessary, we have $\widehat{\zeta}^{k_j+1} \to \bar{\zeta} \in F(\bar{x})$. Taking the limit in (24), we obtain that

$$\left\langle \bar{\zeta} + \sum_{i=1}^{m} \bar{\mu}_i \nabla_y g_i(\bar{x}, \bar{x}), y - \bar{x} \right\rangle \geq 0, \text{ with } \bar{\zeta} \in F(\bar{x}).$$

Because $\bar{\mu}_i \geq 0$ and $g_i(y, \bar{x}) - g_i(\bar{x}, \bar{x}) \geq \langle \nabla_y g_i(\bar{x}, \bar{x}), y - \bar{x} \rangle$ by convexity of $g(\cdot, \bar{x})$, we have that

$$\begin{aligned} \langle \bar{\zeta}, y - \bar{x} \rangle &\geq -\sum_{i=1}^{m} \bar{\mu}_i \langle \nabla_y g_i(\bar{x}, \bar{x}), y - \bar{x} \rangle \\ &\geq -\sum_{i=1}^{m} \bar{\mu}_i (g_i(y, \bar{x}) - g_i(\bar{x}, \bar{x})). \end{aligned} \tag{25}$$

By condition (11) in Theorem 1, $\mu_i^{k_j} g_i(x^{k_j}, x^{k_j}) = 0$ and, hence, $\bar{\mu}_i g_i(\bar{x}, \bar{x}) = 0$. Also, $\bar{\mu}_i \geq 0$ and, since $y \in K(\bar{x})$, $\bar{\mu}_i g_i(y, \bar{x}) \leq 0$. From (25) it follows that $\langle \bar{\zeta}, y - \bar{x} \rangle \geq 0$, which shows that $(\bar{x}, \bar{\zeta}) \in K(\bar{x}) \times F(\bar{x})$ solves (1)-(2), as stated. □

We next discuss an important class of problems with a certain block-separable structure.

4 Block-Separable Approximations

In many applications, including the one considered in Subsection 5.1, the subproblem decision vector $y \in \mathbb{R}^n$ can be split into subvectors, say,

$$y = \prod_{a \in A} y_a \text{ for } y_a \in \mathbb{R}^{n_a} \text{ and } \sum_{a \in A} n_a = n\,,$$

according to decomposable structures observed in the set K_h in (2). Specifically, when

$$K_h = \prod_{a \in A} K_{h_a} \text{ for } K_{h_a} = \{y_a \in \mathbb{R}^{n_a} : h_a(y_a) \leq 0\}\ ,$$

then the VI operator in subproblem (7)

$$F^k(y) = \widehat{F}^k(y) + \Gamma^k(y)\mu^k + Q^k(y - x^k)\,,\ \text{for } \widehat{F}^k(y) \text{ from (6) and } \Gamma^k(y)\mu^k \text{ from (5)},$$

can be further decomposed as the product of VI-operators of smaller dimensions:

$$F^k(y) = \prod_{a \in A} \mathcal{F}_a^k(y_a) \text{ where } y = \prod_{a \in A} y_a\,.$$

This is achieved by means of a Jacobi-like approach, similar to the one in [25]. The process starts rearranging the n components of F^k according to the block structure:

$$F^k(y) = \prod_{a \in A} F_a^k(y) \text{ where } y \in \mathbb{R}^n\,.$$

Then, each block $F_a^k(y)$ is approximated by an operator depending only on the ath subvector $y_a \in \mathbb{R}^{n_a}$, fixing the remaining components to those of a known vector, for instance $x^k \in \mathbb{R}^n$:

$$F_a^k(y) \approx \mathcal{F}_a^k(y_a) = F_a^k(y_a, x_{-a}^k) \quad \text{where } y_a \in \mathbb{R}^{n_a} \text{ and } x_{-a}^k = \prod_{a \neq j \in A} x_j^k \in \mathbb{R}^{n - n_a}\,. \tag{26}$$

With this approximation, (7) amounts to solving separate VI subproblems, each one of dimension n_a:

$$\begin{array}{lll} \text{for } a \in A \quad \text{find } (y_a^{k+1}, z_{s_a}^{k+1}) & \text{such that} & y_a^{k+1} \in K_{h_a} \text{ and } z_{s_a}^{k+1} \in \mathcal{F}_a^k(y_a^{k+1}) \\ & \text{satisfy} & \left\langle z_{s_a}^{k+1}, y_a - y_a^{k+1} \right\rangle \geq 0 \ \text{ for all } y_a \in K_{h_a} \\ & \text{where} & \mathcal{F}_a^k(y_a) \text{ is given in (26).} \end{array} \tag{27}$$

To see that all the theoretical results in Section 3 remain valid for this approxiomation, notice first that evaluating each Jacobi block in (26) at $y_a = x_a^k$ gives the identity

$F_a^k(x_a^k) = \mathcal{F}_a^k(x_a^k)$. Taking the product over all the blocks,

$$F^k(x_a^k) = \mathcal{F}^k(x^k) \text{ where } \mathcal{F}^k(y) = \prod_{a \in A} \mathcal{F}_a^k(y_a)\,. \tag{28}$$

As the gap definition in (9) remains the same, both Lemma 2 and Corollary 3 hold. Regarding the result on asymptotic convergence, a crucial property shown below is that the Jacobi approximation preserves the monotonicity of the original map. As a result, the (regularized) Jacobi approximation $\mathcal{F}^k$ can always be strongly monotone as long as F is monotone. Actually, Jacobi approximations need a weaker setting, referred to as block-wise monotonicity of F. Specifically, monotonicity of the whole operator F^k in Theorem 4 can be replaced by requiring monotonicity of the individual blocks $\mathcal{F}_a^k$. This relaxed assumption can be useful in cases where F is not monotone but has blocks which are monotone when some components are fixed. Block-wise monotonicity occurs naturally, for example, in GNEPs.

To simplify the notation, we illustrate these remarks for the single-valued maps only.

Proposition 5 (Monotonicity properties of the Jacobi approximation)**.** *Consider Jacobi approximations as in (26), defined for a single-valued operator F^k.*

If for all $a \in A$ the Jacobi blocks $\mathcal{F}_a^k(y_a)$ are monotone with respect to $y_a \in \mathbb{R}^{n_a}$, then the full operator $\mathcal{F}^k(y) = \prod_{a \in A} \mathcal{F}_a^k(y_a)$ is monotone with respect to $y \in \mathbb{R}^n$.

Proof. By definition of the full Jacobi approximation, we need to show that

$$\forall y, y' \in \mathbb{R}^n, \quad \left\langle \mathcal{F}^k(y) - \mathcal{F}^k(y'), y - y' \right\rangle = \sum_{a \in A} \left\langle \mathcal{F}_a^k(y_a) - \mathcal{F}_i^k(y'_a), y_a - y'_a \right\rangle \geq 0\,.$$

The result follows, because each term in the right hand side is non negative, by assumption. □

The statements in Theorem 4 remain valid when replacing the VI operator F^k and subproblem (7) by the Jacobi approximation $\mathcal{F}^k$ and (27). Most of the proof remains the same, replacing throughout F^k by $\mathcal{F}^k$ (for instance, using (28) in (21) and (20), yields (22)).

5 Numerical Results

In order to assess the performance of our proposal, we apply Algorithm 1 to two well-known problems formulated as quasi-variational inequalities. The first application, that aims at determining a stable state of equilibrium in an abstract economy, provides a good setting to illustrate the interest of the Jacobi approximations presented in Section 4. In the second application, an instance from [17] called *the moving-set problem*, we examine the impact of employing different approximations in (5) in the VI subproblems (7).

The employed variants of the DW decomposition in each application are compared against the direct solution of (1)-(2), without decomposition. The two approaches are respectively referred to as DW and DIRECT.

All of our runs were done on a computer running Ubuntu 22.04 with AMD Ryzen Threadripper 1950X processors, featuring 16 cores (32 threads) and 64GB of RAM. The codes are available in https://github.com/ManoelJardim/DWQVI/tree/main/DWQVI. The codes were written in Python, calling GAMS [30] and using its Extended Mathematical Programming (EMP) tools. The EMP tool was introduced in [34] to ease the GAMS formulation of specific equilibrium problems and their solution using PATH [35]. We do not assign any parameters for GAMS and use its default values.

5.1 Walrasian Equilibrium Problems

The model introduced in [36], stated here as in [11, 16], is a generalized Nash game involving firms, consumers, and market players. More precisely, there is one firm producing G goods that are bought by C consumers, and a player in charge of finding a price p for the goods that clears the market (having more firms is possible, but not essential for this application). The Walrasian equilibrium prices the goods in a manner that maximizes profit for the firm, utility for the consumers, and clears the market to the best possible extent.

5.1.1 Problem formulation and QVI blockwise structure

On the demand side, letting $i = 1, \ldots, C$, and given a price p, the ith consumer defines the goods to be bought, $x^i \in \mathbb{R}^G$, according to preferences determined by a concave utility function $\mathcal{U}^i(\cdot)$ and taking into account an initial endowment $\mathcal{E}^i_j$ for each good (that impacts on the consumer willingness to buy more of the good). The ith consumer problem is

$$\max_{x^i \geq 0} \left\{ \mathcal{U}^i(x^i) : \langle p, x^i \rangle \leq \langle p, \mathcal{E}^i \rangle \right\} . \tag{29}$$

The firm, denoted here as player $C+1$, decides its production of goods, $x^{C+1} \in \mathbb{R}^G$, by solving the problem

$$\max_{x^{C+1} \geq 0} \left\{ \langle p, x^{C+1} \rangle : \sum_{j=1}^{G} (x_j^{C+1})^2 \leq M \right\} , \tag{30}$$

where the capacity constraint depends on a parameter $M > 0$ and the price $p \in \mathbb{R}^G$ is given.

The equilibrium condition means that the supply meets exactly the demand in every market. Because the economy has G goods and there are endowments, the player in charge of clearing the market looks for (normalized) prices that solve the problem

$$\max_{p \geq 0} \left\{ \left\langle p, \sum_{i=1}^{C} (x^i - \mathcal{E}^i) - x^{C+1} \right\rangle : \sum_{j=1}^{G} p_j = 1 \right\} . \tag{31}$$

The Walrasian equilibrium results from reformulating the generalized Nash game associated with (29), (30), and (31) as a quasi-variational inequality (1)-(2). To this aim, we rename the market-clearing player solving (31) as agent $C+2$ and its decision variable $x^{C+2} = p$, so that the decision variable is $x = \left[x^1 \ \dots \ x^C \ x^{C+1} \ x^{C+2}\right]^T$.

For $x \in \mathbb{R}^{(C+2)G}$, the game in the format (1)-(2), has the operator

$$F(x) = \prod_{i=1}^{C} \left(-\nabla_{x^i} \mathcal{U}^i(x^i)\right) \times \begin{bmatrix} -x^{C+2} \\ \sum_{i=1}^{C} (\mathcal{E}^i - x^i) + x^{C+1} \end{bmatrix}, \tag{32}$$

revealing a separable structure in its first C components. The constraint sets in (2) are

$$\begin{aligned} K_g(x) &= \left\{ y \in \mathbb{R}^{(C+2)G} : g_i(y,x) = \sum_{j=1}^{G} x_j^{C+2}(y_j^i - \mathcal{E}_j^i) \leq 0, \text{ for } 1 \leq i \leq C \right\}, \\ K_h &= \left\{ y \in \mathbb{R}^{(C+2)G} : \begin{bmatrix} -y \\ \sum_{j=1}^{G} (y_j^{C+1})^2 - M \end{bmatrix} \leq 0 \text{ and } \sum_{j=1}^{G} y_j^{C+2} - 1 = 0 \right\}. \end{aligned} \tag{33}$$

Since the easy constraints are uncoupled for the different players, we separate K_h following the structure observed in (32). Accordingly, we first define the sets

$$K_h^a = \left\{ y^a \in \mathbb{R}^G : -y^a \leq 0 \right\}, \quad \text{if } 1 \leq a \leq C.$$

Constraints involving variables with index larger than C are gathered into one set, indexed by D:

$$K_h^D = \left\{ y^D = (y^{C+1}, y^{C+2}) \in \mathbb{R}^G \times \mathbb{R}^G : -y^D \leq 0, \right. \\ \left. \sum_{j=1}^{G} (y_j^{C+1})^2 - M \leq 0, \sum_{j=1}^{G} y_j^{C+2} - 1 = 0 \right\},$$

so that

$$\text{for } A = \{1, \dots, C\} \cup \{D\}, \quad y = \prod_{a \in A} y^a \text{ and } K_h = \prod_{a \in A} K^a.$$

In this notation, the operator in (32) has the blockwise expression

$$F(y) = \prod_{a=1}^{C} F_a(y^a) \times F_D(y) \text{ for } \begin{cases} F_a(y^a) = -\nabla_{y^a} \mathcal{U}^a(y^a) \text{ if } 1 \leq a \leq C, \text{ and} \\ F_D(y) = \begin{bmatrix} -y^{C+2} \\ \sum_{i=1}^{C} (\mathcal{E}^i - y^i) + y^{C+1} \end{bmatrix}. \end{cases}$$

To achieve separability in the VI subproblem operator, we make a Jacobi approximation for F_D, fixing the consumer variables to their value at x^k:

$$F_D^k(y) \approx \mathcal{F}_D^k(y^D) = F_D^k(x_1^k, \dots, x_C^k, y^{C+1}, y^{C+2}).$$

Because of the linear dependence on y for $g(y,x)$ in the current setting, any choice for Γ^k in (5) reduces to the constant approximation. We take the exact option for $\widehat{F}^k$ in (6).

5.1.2 Solvers, data generation, and results

For the DIRECT solver, we follow the recommendations in [34, Section 3] to describe GNEPs using EMP. The combination of EMP and GAMS was also employed in DW to solve all master QVI problems (3) and the VI subproblems (27) for the block corresponding to $a = D$. The remaining VI subproblems (27), the consumers' blocks $a = 1, \dots C$, are simple quadratic programs solved calling the CVXOPT package in Python.

To create random instances for (32)-(33), we follow [11, 16, 17] and consider

$$\text{quadratic concave utilities } \mathcal{U}^i(x^i) = -\frac{1}{2}\left\langle x^i, R^i x^i \right\rangle + \left\langle b^i, x^i \right\rangle \text{ for } i = 1, \dots, C\,.$$

The vector b^i has G normally distributed components ranging between 0 and 10. The positive semidefinite matrix R^i is created as follows. First a random matrix A^i of order G is generated, with elements uniformly distributed in $[-1, 1]$. Then, for $B^i = A^{iT}A^i$, we set $R^i = \frac{10}{\|B^i\|_\infty} B^i$, so that R^i elements belong to $[-10, 10]$ regardless the size of G. Endowments are randomly generated with a uniform distribution in $[0, 10]$. Finally, the firm's capacity M was set at a sufficiently large value to meet the market demand in the large problems (the chosen value for M had none or little impact in the results).

The numerical experiments were designed by varying the number of consumers C and goods G, leading to QVI dimensions $n = (C + 2)G$ ranging between 100 and 50,000. The results are grouped in two different benchmarks, noting that 20 different random instances were run for each considered dimension in all the cases.

In BENCHMARK 1, we fix the number of consumers and vary the number of goods: $(C, G) \in \{20\} \times \{20, 30, 50, 100, 150, 200, 250\}$. Both solvers were run until triggering their respective stopping test. For this benchmark, and as a sanity test, we also checked that the solutions found by DIRECT and DW were numerically identical.

Table 1 CPU time for solvers DIRECT and DW, and number of DW iterations with 140 instances of BENCHMARK 1. In each row, the solver in bold face is the one having the lowest mean time of execution.

n	(C,G)	ratio G/C	DIRECT time (s)		DW time (s)		DW iterations	
			mean	max	mean	max	mean	max
440	(20,20)	1	**0.17**	0.19	1.51	2.78	9.75	16
660	(20,30)	1.5	**0.27**	0.29	1.49	2.28	9.30	12
1,100	(20,50)	2.5	**0.59**	0.65	1.41	2.65	7.50	13
2,200	(20,100)	5	2.26	2.51	**1.21**	2.54	5.45	10
3,300	(20,150)	7.5	5.44	5.91	**1.30**	1.80	4.05	5
4,400	(20,200)	10	10.99	11.87	**1.32**	2.37	4.25	7
5,500	(20,250)	12.5	19.57	20.97	**1.54**	1.94	4.05	5

As seen in the output reported in Table 1, DIRECT performs with almost instantaneous execution times for dimensions up to 1,100. But as the problem size increases, DW becomes competitive and largely surpasses DIRECT. The benchmark confirms the positive impact of decomposition: the sequential DW iterative process, solving small QVI master problems and decoupled VI-subproblems, finds the same solutions as DIRECT with eventually less computational effort. Table 1 also hints at an interesting phenomenon, related to the values reported in the third column. Specifically, DW surpasses DIRECT when the ratio G/C is larger than 2.5. Moreover for $G/C \geq 5$ the number of DW iterations and its mean running time remains practically the same, even though the dimension n grows from 2,200 to 5,500.

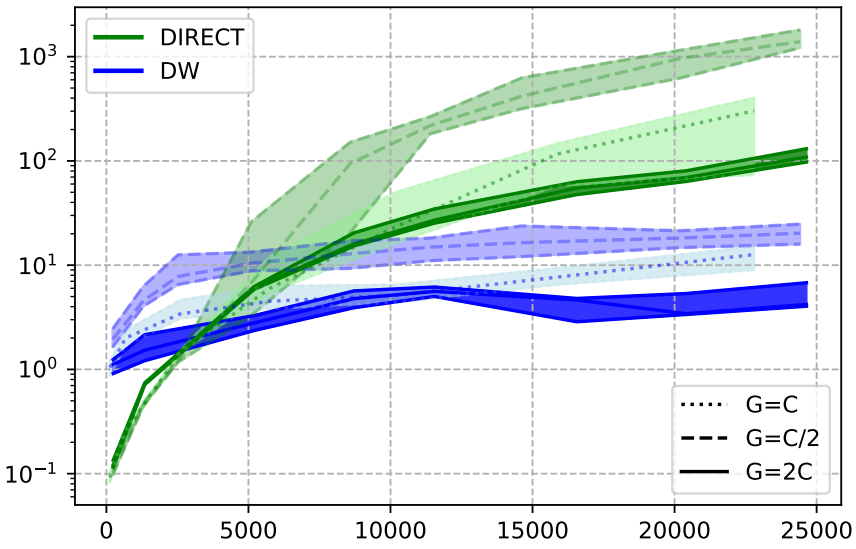

Fig. 2 Time statistics (median and 25%-75% quantiles) for DIRECT and DW, separating the three different ratios between goods and consumers in Table 2. For better illustration, three different economy configurations are plotted separately, in a semilogarithmic scale and barring the last two lines in the table.

In order to include more configurations favorable to the direct solution, the next benchmark varies the ratio C/G in $\{0.5, 1, 2\}$, generating instances with QVI dimension $n \in [120, 40, 400]$. Statistics for the corresponding results are reported graphically in Figure 2, with full details in Table 2.

Figure 2 gives a clear graphical confirmation that, as n increases, DW solving time remains stable while DIRECT's exhibits an exponential growth. Also, the wide shaded areas observed for DIRECT when $G = C$ and $G = C/2$ correspond to the direct solution approach being more sensitive than DW to the ration G/C. In the detailed output in

Table 2, for different configurations of the economy DW performs better than DIRECT as soon as the QVI dimension $n = (C + 2)G$ becomes larger than 5,000.

Table 2 CPU time for solvers DIRECT and DW, and number of DW iterations with 540 instances of BENCHMARK 2. In each row, the solver in bold face is the one having the lowest mean time of execution.

n	(C, G)	ratio G/C	DIRECT time (s)		DW time (s)		DW iterations	
			mean	max	mean	max	mean	max
120	(10, 10)	1	**0.09**	0.14	1.04	1.86	7.80	13
220	(20, 10)	0.5	**0.11**	0.16	2.12	4.06	12.90	22
240	(10, 20)	2	**0.13**	0.15	1.11	2.02	8.30	14
625	(25, 25)	1	**0.25**	0.29	2.10	3.21	11.95	17
1,300	(50, 25)	0.5	**0.47**	0.60	4.99	9.80	19.05	30
1,350	(25, 50)	2	**0.73**	0.78	1.67	3.28	8.05	14
2,520	(70, 35)	0.5	**1.27**	2.00	10.09	21.60	23.55	38
2,600	(50, 50)	1	**1.49**	1.61	3.79	5.69	13.15	18
5,100	(100, 50)	0.5	15.81	65.71	**11.53**	26.89	21.25	36
5,200	(50, 100)	2	6.10	6.84	**2.96**	6.56	8.20	16
5,775	(75, 75)	1	8.18	14.21	**5.00**	8.46	11.75	18
8,580	(130, 65)	0.5	107.90	327.86	**13.29**	23.29	18.95	28
8,710	(65, 130)	2	19.73	45.98	**4.76**	8.08	6.95	11
10,200	(100, 100)	1	84.27	638.98	**5.96**	11.25	10.20	17
11,400	(150, 75)	0.5	260.89	749.89	**15.21**	29.71	18.35	29
11,550	(75, 150)	2	31.73	82.83	**5.74**	9.00	6.70	10
14,620	(170, 85)	0.5	513.31	1,499.38	**17.73**	31.99	18.25	28
15,875	(125, 125)	1	438.68	2,903.90	**7.81**	11.60	9.30	13
16,560	(90, 180)	2	68.37	195.74	**4.18**	8.38	5.40	10
20,200	(200, 100)	0.5	953.97	2,420.30	**18.08**	31.22	15.85	24
20,400	(100, 200)	2	75.97	113.51	**4.34**	9.13	4.90	9
22,800	(150, 150)	1	450.60	2,331.93	**13.01**	24.37	8.15	14
24,420	(220, 110)	0.5	1,638.28	4,765.29	**21.32**	37.76	15.80	24
24,640	(110, 220)	2	156.74	604.68	**5.24**	8.32	4.85	7
30,975	(175, 175)	1	583.40	5,192.05	**9.23**	21.55	6.25	13
31,750	(125, 250)	2	179.81	306.53	**5.31**	5.47	4.00	4
40,400	(200, 200)	1	1,186.40	9,133.32	**10.12**	22.72	6.10	12

For both solvers, the lower the ratio G/C, the harder the problem becomes, because of the number of constraints. In all the considered instances, DW took less than 20 seconds in average, while DIRECT times varied between 0.09 seconds and 30 minutes. As hinted by the shaded areas in Figure 2 and noticeable in the fifth column in Table 2, DIRECT's maximum execution times appear to be highly sensitive to the economy configuration.

5.2 Moving Set Problems

We now turn our attention to problems MovSet3A1, MovSet3B1, MovSet3A2, and MovSet3B2 from [17], see also [16]. In these problems, there is no h constraint and $g(x, y)$ depends quadratically on y.

Given positive dinite matrices $A, R \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n$, and $d \in \mathbb{R}$, the QVI operator in (1) and constraint sets in (2) are

$$F(x) = Ax + b \text{ for } x \in \mathbb{R}^n\,,$$
$$K_g(x) = \{y \in \mathbb{R}^n : g(y,x) = \langle R(y - Bx), y - Bx\rangle - d \leq 0\}, \text{ and } K_h = \mathbb{R}^n\,.$$

As $\nabla_y g(y,x) = 2R(y - Bx)$, the setting is suitable to analyze the impact of different values of ω for $\Gamma_j^k(y)$ in (5). Since F is a linear operator, we take the (exact) approximation $\widehat{F}^k(y) = F(y)$ in (6), which yields, for $\omega^k \in [0,1]$ and $\mu^k \geq 0$, the following VI operator:

$$F^k(y) = Ay + b + \Big((1-\omega^k)2^k(Ry - Bx^k) + \omega^k 2(R-B)x^k\Big)\mu^k\,.$$

As $K_h = \mathbb{R}^n$, in this case subproblems (7) are simple linear systems of equations on y.

The comparison with DIRECT is reported in Table 3, indicating with bold face the fastest solver for each run. For this simple test set, subproblems are solved with no

Table 3 MovSet problems using different ω in Γ^k for DW.

Problem	n	DIRECT (s)	DW (s) $\omega = 1$	DW (s) $\omega = 0$	DW (s) $\omega = 0.5$
MovSet3A1	1,000	51.97	1.71 (11 it.)	**0.74 (5 it.)**	0.83 (6 it.)
MovSet3B1	1,000	50.88	1.64 (11 it.)	**0.72 (5 it.)**	0.90 (6 it.)
MovSet3A2	2,000	285.02	4.60 (12 it.)	**1.90 (5 it.)**	3.00 (7 it.)
MovSet3B2	2,000	284.24	4.44 (11 it.)	**1.75 (5 it.)**	1.92 (6 it.)

effort and it is best to include the full information of the relaxed constraint g ($\omega = 0$ i.e. the free option in (5)).

Declarations

Conflicts of Interest. The authors declare that they have no conflict of interest of any kind related to the manuscript.

Acknowledgements. The second author is supported by CNPq Grant 307509/2023-0 and by PRONEX–Optimization. The third author is supported in part by CNPq Grant 306775/2023-9, by FAPERJ Grant E-26/200.347/2023, and by PRONEX–Optimization.

The authors thank to Professor Paulo J. S. Silva from UNICAMP for facilitating the access to IMECC's computational resources that were necessary for conducting our numerical experiments.

References

[1] Facchinei, F., Pang, J.-S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer Series in Operations Research and Financial Engineering. Springer, New York (2004). https://doi.org/10.1007/b97543

[2] Kanzow, C., Steck, D.: Quasi-variational inequalities in Banach spaces: Theory and augmented Lagrangian methods. SIAM Journal on Optimization **29**(4), 3174–3200 (2019) https://doi.org/10.1137/18m1230475

[3] Bensoussan, A., Goursat, M., Lions, J.L.: Controle impulsionnel et inequations quasi-variationnelles stationnaires. Comptes Rendus de l'Académie des Sciences Paris, Series A **276**, 1279–1284 (1973)

[4] Outrata, J., Kočvara, M., Zowe, J.: Nonsmooth Approach to Optimization Problems with Equilibrium Constraints. Springer, US (1998). https://doi.org/10.1007/978-1-4757-2825-5

[5] Kravchuk, A.S., Neittaanmäki, P.J.: Variational and Quasi-Variational Inequalities in Mechanics. Springer, Netherlands (2007). https://doi.org/10.1007/978-1-4020-6377-0

[6] Scrimali, L.: Quasi-variational inequalities in transportation networks. Mathematical Models and Methods in Applied Sciences **14**(10), 1541–1560 (2004) https://doi.org/10.1142/s0218202504003714

[7] Harker, P.T.: Generalized Nash games and quasi-variational inequalities. European Journal of Operational Research **54**(1), 81–94 (1991) https://doi.org/10.1016/0377-2217(91)90325-p

[8] Aussel, D., Donato, M.B., Milasi, M., Sultana, A.: Existence results for quasi-variational inequalities with applications to Radner equilibrium problems: Resolution through variational inequalities. Set-Valued and Variational Analysis **29**(4), 931–948 (2021) https://doi.org/10.1007/s11228-021-00608-x

[9] Facchinei, F., Kanzow, C.: Generalized Nash equilibrium problems. 4OR **5**(3), 173–210 (2007) https://doi.org/10.1007/s10288-007-0054-4

[10] Fukushima, M.: A class of gap functions for quasi-variational inequality problems. Journal of Industrial and Management Optimization **3**(2), 165–171 (2007) https://doi.org/10.3934/jimo.2007.3.165

[11] Facchinei, F., Kanzow, C.: Penalty methods for the solution of generalized Nash equilibrium problems. SIAM Journal on Optimization **20**(5), 2228–2253 (2010) https://doi.org/10.1137/090749499

[12] Kanzow, C.: On the multiplier-penalty-approach for quasi-variational inequalities. Mathematical Programming **160**(1–2), 33–63 (2016) https://doi.org/10.1007/s10107-015-0973-3

[13] Kanzow, C., Steck, D.: Augmented Lagrangian and exact penalty methods for quasi-variational inequalities. Computational Optimization and Applications **69**(3), 801–824 (2017) https://doi.org/10.1007/s10589-017-9963-0

[14] Hintermüller, M., Rautenberg, C.N.: A sequential minimization technique for elliptic quasi-variational inequalities with gradient constraints. SIAM Journal on Optimization **22**(4), 1224–1257 (2012) https://doi.org/10.1137/110837048

[15] Outrata, J.V., Zowe, J.: A Newton method for a class of quasi-variational inequalities. Computational Optimization and Applications **4**(1), 5–21 (1995) https://doi.org/10.1007/bf01299156

[16] Facchinei, F., Kanzow, C., Sagratella, S.: Solving quasi-variational inequalities via their KKT conditions. Mathematical Programming **144**(1–2), 369–412 (2013) https://doi.org/10.1007/s10107-013-0637-0

[17] Facchinei, F., Kanzow, C., Sagratella, S.: QVILIB: A library of quasi-variational inequality test problems. (2012). https://api.semanticscholar.org/CorpusID:119087660

[18] Facchinei, F., Fischer, A., Piccialli, V.: On generalized Nash games and variational inequalities. Operations Research Letters **35**(2), 159–164 (2007) https://doi.org/10.1016/j.orl.2006.03.004

[19] Kulkarni, A.A., Shanbhag, U.V.: On the variational equilibrium as a refinement of the generalized Nash equilibrium. Automatica **48**(1), 45–55 (2012) https://doi.org/10.1016/j.automatica.2011.09.042

[20] Luna, J.P., Sagastizábal, C., Solodov, M.: In: Kovacevic, R.M., Pflug, G.C., Vespucci, M.T. (eds.) Complementarity and Game-Theoretical Models for Equilibria in Energy Markets: Deterministic and Risk-Averse Formulations, pp. 231–258. Springer, Boston, MA (2013). https://doi.org/10.1007/978-1-4614-9035-7_10

[21] Aussel, D., Sagratella, S.: Sufficient conditions to compute any solution of a quasi-variational inequality via a variational inequality. Mathematical Methods of Operations Research **85**(1), 3–18 (2016) https://doi.org/10.1007/s00186-016-0565-x

[22] Dantzig, G.B., Wolfe, P.: The decomposition algorithm for linear programs. Econometrica **29**(4), 767 (1961) https://doi.org/10.2307/1911818

[23] Fuller, J.D., Chung, W.: Dantzig-Wolfe decomposition of variational inequalities. Computational Economics **25**(4), 303–326 (2005) https://doi.org/10.1007/

s10614-005-2519-x

[24] Chung, W., Fuller, J.D.: Subproblem approximation in Dantzig-Wolfe decomposition of variational inequality models with an application to a multicommodity economic equilibrium model. Operations Research **58**(5), 1318–1327 (2010) https://doi.org/10.1287/opre.1090.0803

[25] Luna, J.P., Sagastizábal, C., Solodov, M.: A class of Dantzig–Wolfe type decomposition methods for variational inequality problems. Mathematical Programming **143**(1–2), 177–209 (2012) https://doi.org/10.1007/s10107-012-0599-7

[26] Luna, J.P., Sagastizábal, C., Solodov, M.: An approximation scheme for a class of risk-averse stochastic equilibrium problems. Math. Program. **157**(2), 451–481 (2016) https://doi.org/10.1007/s10107-016-0988-4

[27] Luna, J.P., Sagastizábal, C., Solodov, M.: A class of Benders decomposition methods for variational inequalities. Comput. Optim. Appl. **76**(3), 935–959 (2020) https://doi.org/10.1007/s10589-019-00157-y

[28] Deride, J., Jofré, A., Wets, R.J.-B.: Solving deterministic and stochastic equilibrium problems via augmented walrasian. Computational Economics **53**(1), 315–342 (2019) https://doi.org/10.1007/s10614-017-9733-1

[29] Borges, P., Sagastizábal, C., Solodov, M.: Decomposition algorithms for some deterministic and two-stage stochastic single-leader multi-follower games. Comput. Optim. Appl. **78**(3), 675–704 (2021) https://doi.org/10.1007/s10589-020-00257-0

[30] GAMS: General Algebraic Modeling System Release 40.3.0. Fairfax, VA, USA (2022). Release Date: Sep 16, 2022. Available for download at https://www.gams.com/download/

[31] Rockafellar, R.T.: Local boundedness of nonlinear, monotone operators. Michigan Mathematical Journal **16**(4) (1969) https://doi.org/10.1307/mmj/1029000324

[32] Burachik, R.S., Iusem, A.N.: Maximal Monotone Operators, pp. 121–159. Springer, Boston, MA (2008). https://doi.org/10.1007/978-0-387-69757-4 . https://doi.org/10.1007/978-0-387-69757-4

[33] Solodov, M.V.: Constraint Qualifications. John Wiley and Sons, Ltd, Chichester, UK (2011). https://doi.org/10.1002/9780470400531.eorms0978

[34] Kim, Y., Ferris, M.C.: Solving equilibrium problems using extended mathematical programming. Mathematical Programming Computation **11**(3), 457–501 (2019) https://doi.org/10.1007/s12532-019-00156-4

[35] Dirkse, S.P., Ferris, M.C.: The PATH solver : A nonmonotone stabilization scheme

for mixed complementarity problems. Optimization Methods and Software **5**, 123–156 (1995)

[36] Arrow, K.J., Debreu, G.: Existence of an equilibrium for a competitive economy. Econometrica **22**(3), 265 (1954) https://doi.org/10.2307/1907353