Mathematical Programming manuscript No.
(will be inserted by the editor)

A class of Dantzig–Wolfe type decomposition methods for variational inequality problems

Juan Pablo Luna · **Claudia Sagastizábal** ·
Mikhail Solodov

Received: date / Accepted: date

Abstract We consider a class of decomposition methods for variational inequalities, which is related to the classical Dantzig–Wolfe decomposition of linear programs. Our approach is rather general, in that it can be used with certain types of set-valued or nonmonotone operators, as well as with various kinds of approximations in the subproblems of the functions and derivatives in the single-valued case. Also, subproblems can be solved approximately. Convergence is established under reasonable assumptions. We also report numerical experiments for computing variational equilibria of the game-theoretical models of electricity markets. Our numerical results illustrate that the decomposition approach allows to solve large-scale problem instances otherwise untractable if the widely used PATH solver is applied directly, without decomposition.

Keywords variational inequality · decomposition · Dantzig–Wolfe decomposition · Josephy–Newton approximation · Jacobi approximation · variational equilibrium.

Mathematics Subject Classification (2000) 90C25 · 90C30 · 65K05.

The first author is supported by FAPERJ scholarship 100.491/2011. The second author is partially supported by Grants CNPq 303840/2011-0, AFOSR FA9550-08-1-0370, NSF DMS 0707205, as well as by PRONEX-Optimization and FAPERJ. The third author is supported in part by CNPq Grant 302637/2011-7, by PRONEX-Optimization and by FAPERJ.

J. P. Luna
IMPA – Instituto de Matemática Pura e Aplicada,
Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil.
E-mail: jluna@impa.br

C. Sagastizábal
CEPEL, Electric Energy Research Center. On leave from INRIA Rocquencourt, France.
IMPA, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro RJ 22460-320, Brazil.
E-mail: sagastiz@impa.br

M. Solodov
IMPA – Instituto de Matemática Pura e Aplicada,
Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil.
E-mail: solodov@impa.br

1 Introduction

Let $F:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a set-valued function from $\mathbb{R}^n$ to the subsets of $\mathbb{R}^n$, and let S_h and S_g be two closed convex sets in $\mathbb{R}^n$. We consider the *variational inequality problem* $\mathrm{VI}(F, S_h \cap S_g)$ [14], which means to find

$$\bar{x} \in S_h \cap S_g \text{ such that } \langle \bar{w}, x-\bar{x}\rangle \geq 0 \text{ for some } \bar{w} \in F(\bar{x}) \text{ and all } x \in S_h \cap S_g. \tag{1}$$

In what follows, we assume that $S_h = \{x : h(x) \leq 0\}$ with $h:\mathbb{R}^n \to \mathbb{R}^q$ being convex and differentiable, and S_g is a generic closed convex set which is easier (in some sense) to handle than the set S_h and the intersection $S_h \cap S_g$. The set S_h is defined by inequality constraints only for simplicity; affine equality constraints can be introduced in our developments without any difficulties. We assume that the operator F is either single-valued and continuous (possibly nonmonotone) or it is maximal monotone, i.e., it is monotone (it holds that $\langle u-v, x-y\rangle \geq 0$ for all $x,y \in \operatorname{dom} F = \{z : F(z) \neq \emptyset\}$ and all $u \in F(x)$, $v \in F(y)$) and its graph is not contained in the graph of any other monotone operator. We also assume that $\mathrm{VI}(F, S_h \cap S_g)$ has a nonempty solution set, and that $S_g \subset \operatorname{int}(\operatorname{dom} F)$. We note that the latter assumption could be more general; we use the stated one for simplicity, as the issue does not seem to be of real importance in a paper devoted to a computational algorithm.

The setting just described suggests trying to deal with the constraint sets S_h and S_g separately, i.e., by some type of decomposition of the problem $\mathrm{VI}(F, S_h \cap S_g)$. Many decomposition techniques (for monotone problems) are explicitly derived from the proximal point method [29,32] for maximal monotone operators, e.g., [41,10,42,44]. Sometimes the relation to the proximal iterates is less direct, e.g., the methods in [4,11,43,17,31], which were nevertheless more recently generalized and interpreted in [37,27] within the hybrid inexact proximal schemes of [39,30]. As some other decomposition methods, we might mention [22] which employs projection and cutting-plane techniques for certain structured problems, matrix splitting for complementarity problems in [6], and the applications of the latter to stochastic complementarity problems in [35]. The methods cited above typically assume monotonicity and, from the beginning, some rather specific structure in the mapping F and/or in the constraints defining the feasible set. In that sense, our setting $\mathrm{VI}(F, S_h \cap S_g)$ and the subsequent developments are more general, as in the single-valued case F is allowed to be nonmonotone and no specific structural assumptions are being made about F or about the constraints. That said, if separable features are present, they can be exploited at the stage of solving the subproblems.

The decomposition approach in this paper is of the type of the Dantzig–Wolfe technique for linear programming [8], which we recall next. Given the affine functions $f:\mathbb{R}^n \to \mathbb{R}$, $h:\mathbb{R}^n \to \mathbb{R}^p$ and $g:\mathbb{R}^n \to \mathbb{R}^q$ (there is no need to define them here explicitly), consider the linear program

$$\min \quad f(x) \quad \text{s.t.} \quad h(x) \leq 0,\ g(x) \leq 0. \tag{2}$$

Note that this problem is equivalent to $\mathrm{VI}(f', S_h \cap S_g)$, where f' is the gradient of f, the set S_h is defined above, and $S_g = \{x : g(x) \leq 0\}$. Assuming that the structure of the problem data is such that linear optimization over the set S_g can be carried out easily (as compared to minimizing over $S_h \cap S_g$), the idea of the Dantzig–Wolfe method is to perform the Lagrangian relaxation of the h-constraints, and to apply the cutting-plane algorithm [2, Sec. 9.3.2] for nonsmooth optimization to the Lagrangian dual

$$\max_{\mu \in \mathbb{R}^q_+} \quad \theta(\mu),$$

where

$$\theta(\mu) = \inf_{x \in S_g} \{ f(x) + \langle h(x), \mu \rangle \}. \tag{3}$$

When set in an iterative framework, Dantzig-Wolfe method means the following. Given the current dual iterate $\mu_M^k \in \mathbb{R}_+^q$, the k-th subproblem consists of computing the value of the dual function $\theta(\mu_M^k)$, which means obtaining a minimizer x_S^{k+1} for the linear program in (3). By construction, $x_S^{k+1} \in S_g$ and we have a subgradient $-h(x_S^{k+1}) \in \partial(-\theta)(\mu_M^k)$ (where $\partial(-\theta)$ denotes the subdifferential of the convex function $-\theta$). For $k \geq 1$, suppose we have computed solutions of the previous subproblems $\{x_S^0, \ldots, x_S^k\} \subset S_g$. Then the k-th master problem replaces the set S_g in the original problem (2) by the approximation $\operatorname{conv}\{x_S^0, \ldots, x_S^k\}$, where $\operatorname{conv} D$ stands for the convex hull of the set D, and solves the following linear program:

$$\begin{cases} \min f(x) \\ x \in \operatorname{conv}\{x_S^0, \ldots, x_S^k\} \\ x \in S_h \end{cases} \quad \Leftrightarrow \quad \begin{cases} \min f\left(\sum_{i=0}^k \alpha_i x_S^i\right) = \sum_{i=0}^k \alpha_i f(x_S^i) \\ \alpha \in \Omega_{k+1} \\ h\left(\sum_{i=0}^k \alpha_i x_S^i\right) = \sum_{i=0}^k \alpha_i h(x_S^i) \leq 0, \end{cases} \tag{4}$$

where $\Omega_{k+1} = \{\alpha \in \mathbb{R}_+^{k+1} : \sum_{j=0}^k \alpha_i = 1\}$ is the unit simplex in $\mathbb{R}^{k+1}$. This gives a solution x_M^k and a multiplier $\mu_M^k \in \mathbb{R}_+^q$ associated to the h-constraint in (4). Then the new dual function value $\theta(\mu_M^k)$ is computed, as well as an associated x_S^{k+1} in (3), and the process is repeated. From the point of view of maximizing the dual function θ, the dual problem of the master problem (4) corresponds to an iteration of the cutting-plane method, i.e., to solving the linear programming formulation of

$$\max_{\mu \in \mathbb{R}_+^q} \theta^k(\mu) := \min_{i=0,\ldots,k} \{ f(x_S^i) + \langle h(x_S^i), \mu \rangle \}.$$

The Dantzig–Wolfe approach for linear programs is particularly effective when h is the coupling constraint without which the minimization in (3) decomposes further according to some favorable (block-separable) structure of g. For some large-scale applications of this type, the resulting method is faster than solving the original linear program (2) directly, e.g., [19,7]. Stabilization techniques of bundle methods [2, Ch. 10] can be used to define stabilized master problems and improve computational performance of the cutting-plane scheme [1,3].

Let us now go back to the variational problem (1). As is well known, it is equivalent to the inclusion $0 \in F(\bar{x}) + \mathcal{N}_{S_h \cap S_g}(\bar{x})$, where by $\mathcal{N}_D(x)$ we denote the normal cone to the convex set D at the point x, i.e., $\mathcal{N}_D(x) = \{v : \langle v, u - x \rangle \leq 0 \; \forall u \in D\}$ if $x \in D$ and $\mathcal{N}_D(x) = \emptyset$ otherwise. Let $\bar{x} \in S_h \cap S_g$. Under appropriate constraint qualification conditions for the sets S_h and S_g (see, e.g., [14, Chapter 3.2] and [38]), it holds that $\mathcal{N}_{S_h \cap S_g}(\bar{x}) = \mathcal{N}_{S_h}(\bar{x}) + \mathcal{N}_{S_g}(\bar{x})$ and $\mathcal{N}_{S_h}(\bar{x}) = \{u : u = [h'(\bar{x})]^\top \mu, \; \mu \in \mathbb{R}_+^q, \; \mu \perp h(\bar{x})\}$, where the notation $u \perp v$ means that $\langle u, v \rangle = 0$. In particular, for any solution $\bar{x}$ of (1), there exists a multiplier $\bar{\mu} \in \mathbb{R}_+^q$ such that

$$0 \in F(\bar{x}) + [h'(\bar{x})]^\top \bar{\mu} + \mathcal{N}_{S_g}(\bar{x}), \quad 0 \leq \bar{\mu} \perp h(\bar{x}) \leq 0.$$

Hence, solving problem (1) is equivalent to finding $(\bar{x}, \bar{\mu})$ such that

$$\begin{cases} (\bar{x}, \bar{\mu}) \in S_h \times \mathbb{R}_+^q, \quad \bar{\mu} \perp h(\bar{x}), \\ \bar{x} \text{ solves } \mathrm{VI}(F(\cdot) + [h'(\cdot)]^\top \bar{\mu}, S_g). \end{cases} \tag{5}$$

A natural extension of the ideas of the Dantzig–Wolfe decomposition to this variational setting is then the following. Using the current multiplier estimate $\mu_M^k \in \mathbb{R}_+^q$ (instead of

the unknown "optimal" $\bar{\mu}$), the k-th subproblem solves a variational inequality with the structure in (5), to obtain a new primal point x_S^{k+1}. In particular, this variational problem is over the simpler set S_g, with the h-constraint dealt with in a manner similar to the Lagrangian relaxation approach. Using solutions of the previous subproblems $\{x_S^0,\dots,x_S^k\}\subset S_g$, the k-th master problem solves a variational inequality with the structure in (1), except that the set S_g therein is approximated by $\operatorname{conv}\{x_S^0,\dots,x_S^k\}$ (recall (4)). This gives a solution x_M^{k+1} and a new multiplier estimate $\mu_M^{k+1}\in\mathbb{R}^q_+$ for the h-constraint, and the process is repeated. Thus, we iteratively generate two sequences of (approximate) solutions of the problems (1) and (5), using at each iteration the solution of one problem to improve the solution of the other. We shall postpone the details and various possible options to be discussed later.

A Dantzig–Wolfe method along these lines had been introduced for (single-valued) variational inequalities in [16,5]. In [16] some restrictive assumptions are employed. For example, F is required to be either strictly monotone or to be a separable combination of a strictly monotone part with a gradient of a differentiable convex function. The subproblems have the specific form $\mathrm{VI}(F(\cdot)+[h'(x_M^k)]^\top\mu_M^k,S_g)$. Also, the solvability of all the subproblems is an assumption. Some of the restrictive assumptions have been alleviated in [5], where also a useful feature of approximating F in the subproblems is introduced. The latter can be helpful in applications where the subproblem $\mathrm{VI}(F(\cdot)+[h'(x_M^k)]^\top\mu_M^k,S_g)$ is not decomposable, but using instead of F a suitable approximation makes it decomposable and thus easier to solve. One possibility considered in [5] is fixing the value of F at the last master solution, i.e., solving $\mathrm{VI}(F(x_M^k)+[h'(x_M^k)]^\top\mu_M^k,S_g)$. The other possibility uses a Jacobi-like approximation, where only some components of x are fixed at their values at x_M^k. In this paper, we shall also consider other approximations, for example of the Josephy–Newton type [20], which approximates $F(\cdot)$ in the smooth single-valued case by $F(x_M^k)+F'(x_M^k)(\cdot-x_M^k)$. We shall also allow combinations of various approximations. In fact, in our numerical results in Section 4, we found that the *combination* of the Newtonian and Jacobi approximations works best for large problems of the structure considered there. In addition, and as compared to [16,5], our framework also allows for approximations to the derivative h'; does not assume solvability of the subproblems; allows for inexact solutions of subproblems; gives an option of generating (cheap) additional cuts by projecting a selection of previous iterates using separation ideas [24,36]; and can handle the general case of F being set-valued.

The rest of the paper is organized as follows. In Section 2 we formally state the algorithm and discuss the approximation options for F and h', inexact solution of subproblems, and other details. Convergence analysis is given in Section 3. Numerical results for computing variational equilibria of game-theoretical models of electricity markets are presented in Section 4. In particular, we show that some specific implementations of our approach make it possible to solve problem instances which are too large to be handled by the widely used PATH solver [9,15] applied directly to the full problem without decomposition. This is also a difference with the methods in [16,5] where the considered examples were solved faster without decomposition than with decomposition.

A few words about our notation are in order. For a closed convex set D and a point x, by $P_D(x)$ we denote the projection of x onto D. Given the set S_h defined above and any polyhedral set D, we say that the (generalized) Slater constraint qualification holds for the set $S_h\cap D$ if there exists $x\in D$ such that $h(x)<0$. The function $F:\mathbb{R}^n\rightrightarrows\mathbb{R}^n$ is strictly monotone if $\langle u-v,x-y\rangle>0$ for all $x,y\in\operatorname{dom}F$ and $u\in F(x)$, $v\in F(y)$ with $x\neq y$; and it is strongly monotone if there exists $c>0$ such that $\langle u-v,x-y\rangle\geq c\|x-y\|^2$ for any choices above. We say that $F:\mathbb{R}^n\rightrightarrows\mathbb{R}^n$ is outer semicontinuous if for any sequences $\{x^k\},\{y^k\}$ such that $\{x^k\}\to\bar{x}$ and $\{y^k\}\to\bar{y}$ with $y^k\in F(x^k)$, it holds that $\bar{y}\in F(\bar{x})$. We say that a family of set-valued functions $\{F^k\}$ is equicontinuous on compact sets if for every compact

set D and every $\varepsilon > 0$ there exists $\delta > 0$ such that for any $x, y \in D$ with $\|x - y\| < \delta$, for every k it holds that $d_H(F^k(x), F^k(y)) < \varepsilon$, where d_H is the Hausdorff distance bewteen the sets defined by

$$d_H(A,B) = \inf\{r > 0 : A \subset B + B(0,r) \text{ and } B \subset A + B(0,r)\}.$$

Note that since we assume that F is either continuous or maximal monotone, it is therefore outer semicontinuous. In addition, since $S_g \subset \operatorname{int}(\operatorname{dom} F)$, it holds that F is locally bounded on S_g [33].

2 The algorithmic framework

In view of (5), having a current multiplier estimate μ_M^k for the h-constraint, perhaps the first natural approach would be to solve the subproblem

$$\mathrm{VI}(F(\cdot) + [h'(\cdot)]^\top \mu_M^k, S_g). \tag{6}$$

This is a valid option indeed, but it may have drawbacks, at least for some types of problem structures. For example, if (6) involves a general nonlinear (and possibly nonmonotone) mapping F, it may prevent us from taking full advantage of some special structure of the set S_g (e.g., S_g may be block-separable). The same comment applies to the nonlinearity of the derivative of h. Another issue is that the set S_g in (6) may be unbounded (even if $S_h \cap S_g$ were bounded), in which case (6) is not guaranteed to have solutions if F is merely continuous/monotone. For these reasons, we shall consider various approximations to F and h' that include (6) itself as an option, possibly regularized by a variable-metric proximal term to induce solvability of subproblems if needed. The algorithm is as follows.

Algorithm 1 (Dantzig-Wolfe Decomposition)

1. Choose $x_S^0 \in S_g \cap S_h$, such that $h(x_S^0) < 0$ if h is not affine. Choose $\mu_M^0 \in \mathbb{R}_+^q$ and $w_M^0 \in F(x_M^0)$. Set $x_M^0 = x_S^0$ and $k := 0$.
2. **The Subproblem:** Choose an approximation $F^k : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ of $F(\cdot)$, an approximation $H^k : \mathbb{R}^n \to \mathbb{R}^{q \times n}$ of $h'(\cdot)$, a possible modification of μ_M^k given by $\mu^k : \mathbb{R}^n \to \mathbb{R}_+^q$, and a positive (semi)definite matrix $Q_k \in \mathbb{R}^{n \times n}$. Find x_S^{k+1}, an approximate solution of the problem
$$\mathrm{VI}(\hat{F}^k, S_g), \tag{7}$$
$$\hat{F}^k(x) = F^k(x) + [H^k(x)]^\top \mu^k(x) + Q_k(x - x_M^k). \tag{8}$$
3. **The Master Problem:** Choose a finite set X^{k+1} containing $\{x_S^0, \ldots, x_S^{k+1}\}$. Find a solution x_M^{k+1} of the problem
$$\mathrm{VI}(F, S_h \cap \operatorname{conv} X^{k+1}), \tag{9}$$
with the associated $w_M^{k+1} \in F(x_M^{k+1})$ and a Lagrange multiplier μ_M^{k+1} associated to the h-constraint.
4. Set $k := k + 1$ and go to Step 2.

Some comments are in order.

In Step 1, choosing a feasible starting point is needed to guarantee that the master problems (9) are feasible for all k. When h is not affine, the role of the condition $h(x_S^0) < 0$ is to ensure that the Slater constraint qualification holds for the master problems (9) for all k, so that there exist Lagrange multipliers associated to the h-constraint in (9). If h is affine, then (9) is a linearly constrained problem and the existence of Lagrange multipliers is automatic. That said, computing a (strictly) feasible starting point may be nontrivial in some applications. For this reason, Section 2.5 below presents a modification of the algorithm in which the h-constraints are relaxed by introducing slack variables, and computing a starting feasible point is required only for the set S_g (recall that this set is assumed to be simple in our context). Master problems (9) are solved introducing simplicial parametrization of the convex hull, similarly to (4) in the case of linear programs.

The options for approximations F^k and H^k in the subproblems, as well as an augmented-Lagrangian type modification μ^k of the multiplier estimate μ_M^k, will be discussed in Section 2.1 below. As for the regularization matrix Q_k, it should generally be taken as zero if F (and then also F^k, for natural choices) is known to be strongly monotone; if strong monotonicity does not hold then Q_k should be positive definite (e.g., a multiple of the identity; but more sophisticated choices may be useful depending on the structure [30]). The notion of acceptable approximate solutions of subproblems is discussed in Section 2.2.

The set X^{k+1} in the master problem contains previous solutions of subproblems, but we could also add additional points. Section 2.3 shows that, at least if the projection onto the simpler set S_g is easy, we can compute explicitly (at negligible computational cost) points that are improvements over the previous iterates in the sense that they are closer to the solution set of (1).

Some remarks concerning reasonable stopping rules for Algorithm 1 will be given in Section 2.4.

2.1 Approximating the data in the subproblems

We next discuss the options for approximating the problem data in the subproblems. Roughly speaking, possible choices range from the simplest ones of taking the fixed values computed at the previous master solution x_M^k, pass through the Newtonian approximation centered at x_M^k, and arrive to taking the functions themselves ("exact approximation"). Furthermore, different options can be combined. For example, in the differentiable case, we can fix some components of the functions at x_M^k and use Newtonian approximations for the other components. In fact, we found such combinations to be the most efficient ones in our numerical results reported in Section 4.

To be deemed admissible, approximating objects must satisfy the following four basic conditions:

$$w_M^k \in F^k(x_M^k) \subset F(x_M^k), \tag{10a}$$

$$F^k(x) + [H^k(x)]^\top \mu^k(x) \text{ is maximal monotone and its domain contains } \operatorname{dom} F, \tag{10b}$$

$$H^k(x_M^k) = h'(x_M^k), \tag{10c}$$

$$\mu^k(x_M^k) = \mu_M^k. \tag{10d}$$

The mapping approximations. As already commented, F^k estimates F near x_M^k. Some examples are:

$$F^k_{\text{const}}(x) = \{w_M^k\}, \tag{11a}$$

$$F^k_{\text{exact}}(x) = F(x), \tag{11b}$$

$$F^k_{\text{N}}(x) = F(x_M^k) + F'(x_M^k)(x - x_M^k) \text{ in the single-valued smooth case,} \tag{11c}$$

where the subscript N above stands for "Newton". Note that all these approximations are (maximal) monotone if F is (maximal) monotone, and that F^k_{const} is maximal monotone regardless of any assumptions.

We would like to emphasize that even if F is nonmonotone, we can always choose a maximal monotone approximation F^k. For example, taking F^k_{const}. Also, for specific applications there may exist other (more sophisticated options) of choosing a monotone approximation F^k for a nonmonotone F. One example will be discussed in the sequel in the context of VI associated to generalized Nash equilibrium problems and in our numerical results in Section 4.

Approximations of the derivative of the h-constraint. Similarly, the function H^k estimates the derivative h' near the point x_M^k, while preserving the monotonicity property (of derivatives of convex functions). Some examples are:

$$H^k_{\text{const}}(x) = h'(x_M^k), \tag{12a}$$

$$H^k_{\text{exact}}(x) = h'(x), \tag{12b}$$

$$H^k_{\text{N}}(x) = h'(x_M^k) + \sum_{i=1}^{q} h_i''(x_M^k)(x - x_M^k). \tag{12c}$$

Note that for all the cases in (12), because of the convexity of h, the following monotonicity property holds:

$$(H^k(y) - H^k(x))(y - x) \geq 0, \quad \text{for all } x, y \in \mathbb{R}^n. \tag{13}$$

Since $\mu_M^k \geq 0$, it then follows that $[H^k(x)]^\top \mu_M^k$ is also monotone. And if F^k is maximal monotone then (10b) holds if we take $\mu^k(x) = \mu_M^k$.

Multiplier modifications. Choices of $\mu^k(x)$ different from μ_M^k are possible if there are linear equality constraints in the definition of the set S_h (formally, in our setting this would correspond to taking two inequalities with opposite signs). Suppose that these equality constraints are given by $\tilde{h}(x) = Ax - a$, where A and a are a matrix and a vector of appropriate dimensions, respectively. We could then use the augmented Lagrangian choice for the corresponding multipliers:

$$\tilde{\mu}^k(x) = \tilde{\mu}_M^k + r_k \tilde{h}(x) = \tilde{\mu}_M^k + r_k A(x - x_M^k),$$

where $r_k > 0$ is the penalty parameter, and we took into account that $x_M^k \in S_h$ and so $\tilde{h}(x_M^k) = Ax_M^k - a = 0$. It can be seen that this choice satisfies the conditions in (10).

Jacobi-type approximations in the block-separable case. To conclude this section, we consider the important special case where S_g is a product of convex sets. That is, $S_g = \prod_{i=1}^m S_{g_i}$ where $S_{g_i} \subset \mathbb{R}^{n_i}$ are closed convex, $i = 1,\dots,m$, $n = \sum_{i=1}^m n_i$. Having chosen the approximations F^k and H^k, the matrix Q_k, and taking $\mu^k(x) = \mu_M^k$, we can write the function $\hat{F}^k$ for the subproblem by blocks:

$$\hat{F}^k(x) = (\hat{F}_1^k(x),\dots,\hat{F}_m^k(x)), \quad \text{with } \hat{F}_i^k(x) \in \mathbb{R}^{n_i},$$

and for every $i = 1,2,\dots,m$, define the Jacobi-like approximations $\hat{F}_{\mathrm{J}_i}^k : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$,

$$\hat{F}_{\mathrm{J}_i}^k(x_i) = \hat{F}_i^k(x_{M_{-i}}^k, x_i),$$

where $(x_{M_{-i}}^k, x_i) = (x_{M_1}^k,\dots,x_{M_{i-1}}^k, x_i, x_{M_{i+1}}^k,\dots,x_{M_m}^k)$ is the vector with all the blocks of variables, except for the i-th, fixed to the master solution. The corresponding estimate $\hat{F}_{\mathrm{J}}^k : \mathbb{R}^n \to \mathbb{R}^n$ for the subproblem is then given by

$$\hat{F}_{\mathrm{J}}^k(x) = (\hat{F}_{\mathrm{J}_1}^k(x_1),\dots,\hat{F}_{\mathrm{J}_m}^k(x_m)).$$

Accordingly, the objects in (8) take the form

$$F^k(x) = (F_1^k(x),\dots,F_m^k(x)), \quad \text{with } F_i^k(x) \in \mathbb{R}^{n_i},$$

$$H^k(x) = [H_1^k(x)|\cdots|H_m^k(x)], \quad \text{with } H_i^k(x) \in \mathbb{R}^q \times \mathbb{R}^{n_i},$$

$$Q_k = [Q_{k_{ij}}], \quad \text{with } Q_{k_{ij}} \in \mathbb{R}^{n_i} \times \mathbb{R}^{n_j}.$$

And, for each $i = 1,2,\dots,m$, we define the Jacobi-like approximations $F_{\mathrm{J}_i}^k : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ and $H_{\mathrm{J}_i}^k : \mathbb{R}^{n_i} \to \mathbb{R}^q \times \mathbb{R}^{n_i}$ by

$$F_{\mathrm{J}_i}^k(x_i) = F_i^k(x_{M_{-i}}^k, x_i), \quad H_{\mathrm{J}_i}^k(x_i) = H_i^k(x_{M_{-i}}^k, x_i).$$

It is easy to see that

$$\hat{F}_{\mathrm{J}_i}^k(x_i) = F_{\mathrm{J}_i}^k(x_i) + [H_{\mathrm{J}_i}^k(x_i)]^\top \mu_M^k + Q_{k_{ii}}(x_i - x_{M_i}^k),$$

and thus

$$\hat{F}_{\mathrm{J}}^k(x) = F_{\mathrm{J}}^k(x) + [H_{\mathrm{J}}^k(x)]^\top \mu_M^k + Q_k^{\mathrm{J}}(x - x_M^k),$$

where $F_{\mathrm{J}}^k(x) = (F_{\mathrm{J}_1}^k(x_1),\dots,F_{\mathrm{J}_m}^k(x_m))$, $H_{\mathrm{J}}^k(x) = [H_{\mathrm{J}_1}^k(x_1)|\cdots|H_{\mathrm{J}_m}^k(x_m)]$ and $Q_k^{\mathrm{J}} = \mathrm{diag}(Q_{k_{11}}, Q_{k_{22}},\dots,Q_{k_{mm}})$. The functions F_{J}^k and $H_{\mathrm{J}}^k(x)$ satisfy all the properties in (10). Moreover, since for every $i = 1,2,\dots,m$ and $x_i, y_i \in \mathbb{R}^{n_i}$ it holds that

$$\langle \hat{F}_{\mathrm{J}_i}^k(y_i) - \hat{F}_{\mathrm{J}_i}^k(x_i), y_i - x_i\rangle = \langle \hat{F}^k(x_{M_{-i}}^k, y_i) - \hat{F}^k(x_{M_{-i}}^k, x_i), (x_{M_{-i}}^k, y_i) - (x_{M_{-i}}^k, x_i)\rangle,$$

it follows that if $\hat{F}^k$ were monotone, strictly monotone or strongly monotone, then $\hat{F}_{\mathrm{J}_i}^k$ would inherit the same property; and therefore so would $\hat{F}_{\mathrm{J}}^k$. We again comment that in some applications (for example, VI associated to generalized Nash equilibrium problems, see Section 4) the full function $F(x) = (F_1(x),\dots,F_n(x))$ can be nonmonotone but each component F_i is monotone in the variable x_i. So, even if the approximations F_{N}^k and F_{exact}^k could be nonmonotone, the approximations $F_{\text{N-J}}^k$ (where "N-J" stands for "Newton-Jacobi") and $F_{\text{exact-Jobi}}^k$ are monotone in that case.

As we found $F^k_{\text{N-J}}$ particularly useful in our numerical experiments, we shall next state it formally. By the definition above, we have $F^k_{\text{N}_i}(x) = F_i(x^k_M) + F_i'(x^k_M)(x - x^k_M)$, and so

$$\begin{aligned} F^k_{\text{N-J}_i}(x_i) &:= F^k_{\text{N}_i}(x^k_{M_{-i}}, x_i) \\ &= F_i(x^k_M) + F_i'(x^k_M)((x^k_{M_{-i}}, x_i) - (x^k_{M_{-i}}, x^k_{M_i})) \\ &= F_i(x^k_M) + F_i'(x^k_M)((0, x_i - x^k_{M_i})) \\ &= F_i(x^k_M) + \nabla_{x_i} F_i(x^k_M)(x_i - x^k_{M_i}). \end{aligned}$$

Then if $F_i(x)$ is monotone in the variable x_i, we have that $F^k_{\text{N-J}_i}(x_i)$ is monotone and so is $F^k_{\text{N-J}}(x)$.

The motivation for the Jacobi approach is that we can take advantage of the separable structure of S_g when solving the subproblems even when F is not separable. Specifically, it can be seen that x^{k+1}_S solves the subproblem $\text{VI}(\hat{F}^k_{\text{J}}, S_g)$ if and only if the components $(x^{k+1}_S)_i$ solve $\text{VI}(\hat{F}^k_{\text{J}_i}, S_{g_i})$, $i = 1, \ldots, m$. Thus, the subproblems in Algorithm 1 decompose according to the structure of S_g. Clearly, such decomposition is also achieved for $F^k_{\text{N-J}}(x)$.

2.2 Inexact solution of subproblems

By approximate solution of subproblem (7) we mean computing some

$$\begin{aligned} x^{k+1}_S \in S_g \text{ such that } & \langle v^{k+1}_S + e^k, y - x^{k+1}_S \rangle \geq 0 \\ & \text{for some } v^{k+1}_S \in \hat{F}^k(x^{k+1}_S) \text{ and all } y \in S_g, \end{aligned} \tag{14}$$

where $e^k \in \mathbb{R}^n$ is the error term accounting for inexactness. This definition of approximate solutions of variational problems was also employed, e.g., in [28,36]. In our convergence analysis, we shall use the following two approaches to controlling inexactness. One is the "relative-error" type:

$$\langle e^k, x^k_M - x^{k+1}_S \rangle \leq \sigma \langle Q_k(x^k_M - x^{k+1}_S), x^k_M - x^{k+1}_S \rangle, \quad \sigma \in [0,1), \tag{15}$$

or its stronger version

$$\|e^k\| \|x^k_M - x^{k+1}_S\| \leq \sigma \langle Q_k(x^k_M - x^{k+1}_S), x^k_M - x^{k+1}_S \rangle, \quad \sigma \in [0,1). \tag{16}$$

The second rule is the "asymptotically exact" type:

$$e^k \to 0 \quad \text{as} \quad k \to \infty. \tag{17}$$

The first rule is more constructive, as it essentially means that the *relative* error (the ratio between the size of the error term e^k and the size of the step $x^k_M - x^{k+1}_S$) in solving the subproblems needs to be small enough but can be fixed by the value of the parameter σ (which need not tend to zero); see [39,30] and references therein for discussions of the advantages of this relative-error approach. That said, verifying (15) clearly requires the explicit knowledge of e^k in (14). Below we explain how e^k can be constructed and the conditions (14) and (15) checked explicitly in the case of continuous $\hat{F}^k$, if we have access to the iterates of the method applied to solve $\text{VI}(\hat{F}^k, S_g)$. Of course, the latter is not the case when a "black-box" solver is used. In that sense, an advantage of the "asymptotical exactness" rule (17) is that it can be argued that in this case the explicit knowledge of e^k in (14) is not necessary. The algorithm used to solve subproblems (7) can be truncated according to any suitable internal

criteria, provided the precision is progressively tightened along the iterations of the outer Algorithm 1. This would generate, at each step, some unknown error term e^k in (14). But as long as the inexactness in solving the subproblems (however it is measured) asymptotically vanishes, it seems valid that the error written in any other form, for example (14), must also tend to zero.

Suppose now that $\hat{F}^k$ is continuous (single-valued) and strongly monotone (recall that a monotone approximation F^k of F always exists even if F is nonmonotone, and strong monotonicity of $\hat{F}^k$ can be induced by adding the proximal regularization with Q_k positive definite when needed). Let an Algorithm A (any suitable algorithm for solving $\mathrm{VI}(\hat{F}^k, S_g)$) generate a sequence $\{y^{k,i}\}$ which, if continued infinitely, is known to converge to the exact solution $\bar{x}_S^{k+1}$ of the subproblem $\mathrm{VI}(\hat{F}^k, S_g)$) as $i \to \infty$. This solution is unique, because $\hat{F}^k$ is strongly monotone. It holds that

$$\bar{x}_S^{k+1} = P_{S_g}(\bar{x}_S^{k+1} - \hat{F}^k(\bar{x}_S^{k+1})).$$

Define the auxiliary sequences $\{z^{k,i}\}$ and $\{e^{k,i}\}$ by

$$z^{k,i} = P_{S_g}(y^{k,i} - \hat{F}^k(y^{k,i})),$$

$$e^{k,i} = (z^{k,i} - \hat{F}^k(z^{k,i})) - (y^{k,i} - \hat{F}^k(y^{k,i})).$$

Since, by continuity of $\hat{F}^k$ and of the projection operator, $\{z^{k,i}\}$ also converges to $\bar{x}_S^{k+1}$ as $i \to \infty$, it holds that

$$\lim_{i\to\infty} e^{k,i} = 0.$$

Observe now that

$$z^{k,i} = P_{S_g}(y^{k,i} - \hat{F}^k(y^{k,i})) = P_{S_g}(z^{k,i} - (\hat{F}^k(z^{k,i}) + e^{k,i})),$$

which means that, at each iteration of Algorithm A, $z^{k,i}$ solves the problem $\mathrm{VI}(\hat{F}^k + e^{k,i}, S_g)$. In other words, the condition (14) holds at every iteration i for $x_S^{k+1} = z^{k,i}$ and $e^k = e^{k,i}$ with the known $e^{k,i}$ defined above. Since $\{e^{k,i}\} \to 0$ as $i \to \infty$, for any reasonable criterion of measuring approximations Algorithm A would yield in a finite number of iterations an approximate solution $x_S^{k+1} = z^{k,i}$ for the subproblem $\mathrm{VI}(\hat{F}^k, S_g)$ with the known, and thus controllable, error $e^k = e^{k,i}$.

The only computational issue with the presented construction is the projection onto S_g to construct the auxiliary points $z^{k,i}$. However, this projection can be explicit for some problems (e.g., onto a box). Also, it may be already computed by Algorithm A in the course of its iterative procedure anyhow. For example, one of the most natural stopping conditions for $\mathrm{VI}(\hat{F}^k, S_g)$ is precisely to check whether $\|y^{k,i} - z^{k,i}\|$ is small (this is the so-called natural residual of VI [14, Chapter 1.5]; if the natural residual is zero then $y^{k,i} = z^{k,i} = \bar{x}_S^{k+1}$ is the exact solution). In particular, most (if not all) projection methods for VIs (see, e.g., [14, Chapter 12.1], [40]) compute the right-hand side in the definition of $z^{k,i}$ as part of the iterates update (perhaps scaled with a stepsize, but this can be easily accounted for) and/or compute the natural residual $(y^{k,i} - z^{k,i})$ for the stopping test. Thus, within projection methods, $z^{k,i}$ and then $e^{k,i}$ are immediately available.

2.3 Managing the feasible set of the master problem

The basic choice is to take $X^{k+1} = \{x_S^0, \dots, x_S^{k+1}\}$. As already mentioned, to ensure feasibility of the master problems (9) it should hold that $x_M^0 = x_S^0 \in S_h \cap S_g$, and for the existence of Lagrange multipliers $h(x_S^0) < 0$ if h is not affine.

However, when F is monotone and the projection onto S_g is cheap (and this is indeed the case for many applications of interest), we can generate at negligible computational cost some additional "improved" points that are closer to the solution set than the past iterates. This procedure is based on separation/projection ideas, e.g., [24,36].

In this strategy we explicitly state that $X^k \subset X^{k+1}$ for all k, i.e., no points are ever deleted from the feasible set of the previous master problem; points can only be added. Then since x_M^k solves $\mathrm{VI}(F, S_h \cap \operatorname{conv} X^k)$ and $x_M^j \in S_h \cap \operatorname{conv} X^k$ for $j \le k$ (since $X^j \subset X^{j+1}$), we have that for the associated $w_M^k \in F(x_M^k)$ it holds that

$$\langle w_M^k, x_M^j - x_M^k \rangle \ge 0 \quad \text{for} \quad j = 1, \dots, k. \tag{18}$$

On the other hand, if $\bar{x}$ is any solution of $\mathrm{VI}(F, S_h \cap S_g)$, since $x_M^k \in S_h \cap S_g$ it holds that $\langle \bar{w}, x_M^k - \bar{x} \rangle \ge 0$ where $\bar{w} \in F(\bar{x})$. Then monotonicity of F (actually, the weaker pseudo-monotonicity property is enough here) implies that $\langle w_M^k, x_M^k - \bar{x} \rangle \ge 0$. Hence, for every k, the solution set of $\mathrm{VI}(F, S_h \cap S_g)$ lies in the halfspace

$$\{x : \langle w_M^k, x - x_M^k \rangle \le 0\}.$$

Thus, in view of (18), all the previous master problem solutions are separated from the solution set of $\mathrm{VI}(F, S_h \cap S_g)$ by the hyperplane $\{x : \langle w_M^k, x - x_M^k \rangle = 0\}$. In fact, as there seem to be no reasons for the inequality (18) to hold as equality, the separation should be expected to be strict for most points. It is then clear that projecting onto the separating hyperplane (can also be with under- or over-relaxation), would move the previous iterates closer to the solution set, thus giving better approximations to the solution [36].

In addition, previous solutions of subproblems could be considered too, i.e., the points with the property

$$\langle w_M^k, x_S^j - x_M^k \rangle \ge 0 \quad \text{for} \quad j = 1, \dots, k \text{ such that } x_S^j \in S_h.$$

If there are such points then they can be projected/improved also. That said, since $x_S^j \in S_h$ need not hold in general, the existence of candidates to project of this kind is not a given (unlike the case with the previous master problem solutions for which the separation property always holds).

Summarizing, we can choose any subset

$$Z^k \subset \{z \in X^k \cup \{x_M^1, \dots, x_M^k\} : \langle w_M^k, z - x_M^k \rangle > 0\}$$

and define

$$X^{k+1} = \{x_S^0, \dots, x_S^{k+1}\} \cup \left\{ P_{S_g}\left[z - \beta_z \frac{\langle w_M^k, z - x_M^k \rangle}{\|w_M^k\|^2} w_M^k \right] : z \in Z^k \right\},$$

where $\beta_z \in (0,2)$ is over/under relaxation parameter ($\beta_z = 1$ corresponds to the projection onto the separating hyperplane). See [36] for formal justifications.

2.4 Stopping conditions

One reasonable stopping criterion for Algorithm 1 is based on monitoring, after solving the subproblem $\mathrm{VI}(\hat{F}^k, K_g)$, the quantity

$$\Delta_k = \langle w_M^k + [h'(x_M^k)]^\top \mu_M^k, x_S^{k+1} - x_M^k \rangle. \tag{19}$$

(This quantity had also been employed in [16,5].)

The motivation for (19) comes from the stopping test of the cutting-plane algorithm for maximizing the dual function (3) in the original Dantzig–Wolfe method for the linear program (2) (see [16, Theorem 7]). For this problem, using the linearity of the data and the fact that $\mu_M^k \perp h(x_M^k)$ (since these solve (4)), we have that

$$\begin{aligned}
\Delta_k &= \langle f'(x_M^k) + h'(x_M^k)^\top \mu_M^k, x_S^{k+1} - x_M^k \rangle \\
&= f(x_S^{k+1}) - f(x_M^k) + \langle \mu_M^k, h(x_S^{k+1}) - h(x_M^k) \rangle \\
&= f(x_S^{k+1}) + \langle \mu_M^k, h(x_S^{k+1}) \rangle - f(x_M^k) \\
&= \theta(\mu_M^k) - \theta^k(\mu_M^k) \le 0,
\end{aligned}$$

i.e., Δ_k measures how well the dual function θ is approximated by its cutting-plane model θ^k at the current dual iterate μ_M^k. It is standard to stop the cutting-plane method when Δ_k becomes small enough [2, Sec. 9.3.2].

Let us now go back to the variational setting. Suppose x_S^{k+1} is an inexact solution of the suproblem $\mathrm{VI}(\hat{F}^k, K_g)$ in the sense of (14). Since $x_M^k \in S_g$, it then holds that

$$\langle v_S^{k+1}, x_S^{k+1} - x_M^k \rangle \le \langle e^k, x_M^k - x_S^{k+1} \rangle.$$

We can write

$$v_S^{k+1} = u_S^{k+1} + Q_k(x_S^{k+1} - x_M^k), \text{ where } u_S^{k+1} \in F^k(x_S^{k+1}) + [H^k(x_S^{k+1})]^\top \mu^k(x_S^{k+1}).$$

Then, for the inexactness criterion (15), we have that

$$\begin{aligned}
\langle u_S^{k+1}, x_S^{k+1} - x_M^k \rangle &\le \langle e^k, x_M^k - x_S^{k+1} \rangle - \langle Q_k(x_S^{k+1} - x_M^k), x_S^{k+1} - x_M^k \rangle \\
&\le -(1-\sigma) \langle Q_k(x_S^{k+1} - x_M^k), x_S^{k+1} - x_M^k \rangle.
\end{aligned}$$

Now, since $F^k(x) + [H^k(x)]^\top \mu^k(x)$ is monotone and $w_M^k + [h'(x_M^k)]^\top \mu_M^k \in F^k(x_M^k) + [H^k(x_M^k)]^\top \mu^k(x_M^k)$, it holds that

$$\begin{aligned}
\Delta_k &= \langle w_M^k + [h'(x_M^k)]^\top \mu_M^k, x_S^{k+1} - x_M^k \rangle \\
&\le \langle u_S^{k+1}, x_S^{k+1} - x_M^k \rangle \\
&\le -(1-\sigma) \langle Q_k(x_S^{k+1} - x_M^k), x_S^{k+1} - x_M^k \rangle \le 0.
\end{aligned} \tag{20}$$

If Q_k is positive definite, then $\Delta_k = 0$ implies $x_S^{k+1} = x_M^k$, and the latter point is a solution of $\mathrm{VI}(F, S_h \cap S_g)$ (see Proposition 2 below, which also deals with case when Q_k may be positive semidefinite if $\hat{F}^k$ is strictly monotone). A value of Δ_k close to zero means that the difference between the points x_S^{k+1} and x_M^k is small, which justifies the stopping test based on Δ_k.

2.5 Relaxing the constraints in the master problem

We now consider the option of relaxing the h-constraints by introducing slack variables. This feature can be useful when computing a feasible starting point in $S_h \cap S_g$ is nontrivial. A starting point in S_g is still needed, but recall that it is assumed to be a simple set in our context. A similar technique had been mentioned in [16,5], although without theoretical analysis.

Suppose that at an iteration $k \geq 0$ we have a finite subset X^{k+1} of S_g containing the subproblems solutions $\{x_S^0, \ldots, x_S^{k+1}\}$. We define the relaxed master feasible set

$$D_k = \{(x,z) \in \operatorname{conv} X^{k+1} \times \mathbb{R}^q : h(x) \leq z\},$$

and the function $F_M^k : \mathbb{R}^n \times \mathbb{R}^q \rightrightarrows \mathbb{R}^n \times \mathbb{R}^q$ by

$$F_M^k(x,z) = F(x) \times \{\zeta_k z\},$$

where $\zeta_k > 0$ is a scalar parameter. Then the relaxed master problem consists of solving

$$\mathrm{VI}(F_M^k, D_k). \tag{21}$$

Note that the set D_k is always nonempty and satisfies the Slater constraint qualification automatically. Also, F_M^k is (strongly) monotone, if so is F.

The new algorithm is given below.

Algorithm 2 (Relaxed Dantzig-Wolfe Decomposition)

1. Choose $x_M^0 \in S_g$, $w_M^0 \in F(x_M^0)$ and $\mu_M^0 \in \mathbb{R}_+^q$. Set $x_M^0 = x_S^0$ and $k := 0$.
2. **The Subproblem:** Choose the function $\hat{F}^k$ as in Algorithm 1 and find x_S^{k+1}, a solution of the problem $\mathrm{VI}(\hat{F}^k, S_g)$.
3. **The Master Problem:** Choose the set X^{k+1} as in Algorithm 1 and the parameter $\zeta_k > 0$. Find a solution (x_M^{k+1}, z^{k+1}) of the problem (21), with the associated $w_M^{k+1} \in F(x_M^{k+1})$ and a Lagrange multiplier μ_M^{k+1} associated to the h-constraint.
4. Set $k := k+1$ and go to Step 2.

Note that solutions of the subproblems and of the master problems belong to the set S_g, but not necessarily to S_h. In Proposition 3 below, we shall see that the z-component of the master solution is actually uniquely defined and is of the form $z^{k+1} = \mu_M^{k+1}/\zeta_k$.

3 Convergence analysis

We first formalize the arguments that show that the algorithm is well-defined, i.e., that all the subproblems and all the master problems have solutions.

As discussed above, we can always choose F^k to be maximal monotone (even if F is not), so that $\hat{F}^k$ would be maximal monotone with its domain containing the domain of F. Moreover, $\hat{F}^k$ can always be made strongly monotone by taking the matrix Q_k positive definite when needed. Then maximal monotonicity and strong monotonicity of $\hat{F}^k$ ensure the existence of the unique solution to subproblem $\mathrm{VI}(\hat{F}^k, S_g)$ (by [34, Theorem 5]). In addition, as already discussed above, our choice of the starting points in both Algorithm 1 and Algorithm 2 ensures that all the master problems are always feasible and satisfy constraint qualifications [38], so that there exist Lagrange multipliers associated to the solutions (if

any). Now, the master problems are variational inequalities either with a continuous mapping or with a maximal monotone one over nonempty compact feasible sets. Solutions to this type of problems exist, by [14, Corollary 2.2.5] and [34, Theorem 5], respectively.

We start our convergence analysis by establishing some key properties of the master problems solutions in Algorithm 1, extending [16, Thm. 4.5] as well as some results in the online appendix of [5].

Proposition 1 *For a solution x_M^{k+1} of the master problem $VI(F, S_h \cap conv X^{k+1})$ in Algorithm 1, the following assertions hold:*

1. *If h is affine or if there exists $\hat{x} \in conv X^{k+1}$ such that $h(\hat{x}) < 0$, then there exists a Lagrange multiplier $\mu_M^{k+1} \in \mathbb{R}_+^q$ associated to the h-constraint.*
2. *For any such multiplier μ_M^{k+1} it holds that x_M^{k+1} solves $VI(F(\cdot) + [h'(\cdot)]^\top \mu_M^{k+1}, conv X^{k+1})$.*
3. *If for any $x \in S_g$ and any $v \in [F(x_M^{k+1}) + [h'(x_M^{k+1})]^\top \mu_M^{k+1}] \cap [-\mathcal{N}_{conv X^{k+1}}(x_M^{k+1})]$ it holds that $\langle v, x - x_M^{k+1} \rangle < 0$ then $x \notin conv X^{k+1}$.*
4. *If x_M^{k+1} is a solution of $VI(F(\cdot) + [h'(\cdot)]^\top \mu_M^{k+1}, S_g)$ then x_M^{k+1} solves $VI(F, S_g \cap S_h)$.*
5. *On the next iteration ($k := k+1$), if x_M^k is a solution of the subproblem $VI(\hat{F}^k, S_g)$ then x_M^k solves $VI(F, S_g \cap S_h)$.*

Proof Since $\operatorname{conv} X^{k+1}$ is a polyhedral set, the linearity of h in the first case or the Slater constraint qualification in the second case guarantee that

$$\mathcal{N}_{S_h \cap \operatorname{conv} X^{k+1}}(x_M^{k+1}) = \mathcal{N}_{S_h}(x_M^{k+1}) + \mathcal{N}_{\operatorname{conv} X^{k+1}}(x_M^{k+1})$$

and

$$\mathcal{N}_{S_h}(x_M^{k+1}) = \{[h'(x_M^{k+1})]^\top \mu : \mu \in \mathbb{R}_+^q, \mu \perp h(x_M^{k+1})\}.$$

Then, since x_M^{k+1} solves $\mathrm{VI}(F, S_h \cap \operatorname{conv} X^{k+1})$, we have that

$$\begin{aligned} 0 &\in F(x_M^{k+1}) + \mathcal{N}_{S_h \cap \operatorname{conv} X^{k+1}}(x_M^{k+1}) \\ &= F(x_M^{k+1}) + \{[h'(x_M^{k+1})]^\top \mu : \mu \in \mathbb{R}_+^q, \mu \perp h(x_M^{k+1})\} + \mathcal{N}_{\operatorname{conv} X^{k+1}}(x_M^{k+1}), \end{aligned}$$

which means the existence of the multiplier μ_M^{k+1} in question.

In particular, it then holds that

$$0 \in F(x_M^{k+1}) + [h'(x_M^{k+1})]^\top \mu_M^{k+1} + \mathcal{N}_{\operatorname{conv} X^{k+1}}(x_M^{k+1}),$$

which means that x_M^{k+1} solves $\mathrm{VI}(F(\cdot) + [h'(\cdot)]^\top \mu_M^{k+1}, \operatorname{conv} X^{k+1})$, as claimed.

Concerning the third assertion, note that any v in question can serve as an element associated to x_M^{k+1} which verifies that the latter is a solution of $\mathrm{VI}(F(\cdot) + [h'(\cdot)]^\top \mu_M^{k+1}, \operatorname{conv} X^{k+1})$. In other words, $\langle v, x - x_M^{k+1} \rangle \geq 0$ for all $x \in \operatorname{conv} X^{k+1}$. Thus, if this inequality does not hold for some $x \in S_g$, it must be the case that $x \notin \operatorname{conv} X^{k+1}$.

Suppose now that x_M^{k+1} solves $\mathrm{VI}(F + [h'(\cdot)]^\top \mu_M^{k+1}, S_g)$, i.e.,

$$0 \in F(x_M^{k+1}) + [h'(x_M^{k+1})]^\top \mu_M^{k+1} + \mathcal{N}_{S_g}(x_M^{k+1}).$$

Since μ_M^{k+1} is a Lagrange multiplier associated to the h-constraint, we have

$$[h'(x_M^{k+1})]^\top \mu_M^{k+1} \in \mathcal{N}_{S_h}(x_M^{k+1}),$$

and hence,

$$0 \in F(x_M^{k+1}) + \mathcal{N}_{S_h}(x_M^{k+1}) + \mathcal{N}_{S_g}(x_M^{k+1}) \subset F(x_M^{k+1}) + \mathcal{N}_{S_h \cap S_g}(x_M^{k+1}),$$

which establishes the fourth assertion.

Finally, since by (10) it holds that

$$\hat{F}^k(x_M^k) \subset F(x_M^k) + [h'(x_M^k)]^\top \mu_M^k,$$

if x_M^k solves the subproblem $\mathrm{VI}(\hat{F}^k, S_g)$, then the previous item implies that it solves our problem $\mathrm{VI}(F, S_g \cap S_h)$. □

Note that, by the third item of Proposition 1, it follows that for the gap defined in (19) whenever $\Delta_k < 0$ we have that $x_S^{k+1} \notin \operatorname{conv} X^k$. Thus, as long as $\Delta_k < 0$, the feasible set of the master problem keeps growing, improving the approximation of the set S_g. If the subproblems are solved exactly then $x_S^{k+1} \in \operatorname{conv} X^k$ for some k implies that $\Delta_k \geq 0$ (actually $\Delta_k = 0$). Then, if $\hat{F}^k$ is at least strictly monotone (which can always be ensured by taking the matrix Q_k positive definite if needed) it holds that $x_M^k = x_S^{k+1}$ is a solution of the problem $\mathrm{VI}(F, S_g \cap S_h)$ (see also [16, Thm. 6] and[5] for some related statements).

Proposition 2 *Let $\hat{F}^k$ be strictly monotone (e.g., Q_k is positive definite). Suppose that in Algorithm 1 it holds that $\Delta_k \geq \langle e^k, x_M^k - x_S^{k+1} \rangle$ for some iteration index k. Then $\Delta_k = 0$ and $x_M^k = x_S^{k+1}$ solves $VI(F, S_g \cap S_h)$.*

Proof Since x_S^{k+1} is an approximate solution, with error e^k, of the subproblem $\mathrm{VI}(\hat{F}^k, S_g)$ in the sense of (14), and since $x_M^k \in S_g$, for the associated $v_S^{k+1} \in \hat{F}^k(x_S^{k+1})$ it holds that

$$\langle e^k, x_M^k - x_S^{k+1} \rangle \geq \langle v_S^{k+1}, x_S^{k+1} - x_M^k \rangle. \tag{22}$$

Then, by the definition of Δ_k in (19), by the monotonicity of $\hat{F}^k$, and by the fact that $w_M^k + [h'(x_M^k)]^\top \mu_M^k \in \hat{F}^k(x_M^k)$, it holds that

$$\begin{aligned} \Delta_k &= \langle w_M^k + [h'(x_M^k)]^\top \mu_M^k, x_S^{k+1} - x_M^k \rangle \\ &\leq \langle v_S^{k+1}, x_S^{k+1} - x_M^k \rangle \\ &\leq \langle e^k, x_M^k - x_S^{k+1} \rangle, \end{aligned} \tag{23}$$

where (22) was also used. But then the assumption that $\Delta_k \geq \langle e^k, x_M^k - x_S^{k+1} \rangle$ implies that

$$\Delta_k = \langle e^k, x_M^k - x_S^{k+1} \rangle.$$

Then by substituting the expression for Δ_k into the left-hand side of (22) we obtain that

$$\langle w_M^k + [h_2'(x_M^k)]^\top \mu_M^k - v_S^{k+1}, x_S^{k+1} - x_M^k \rangle \geq 0.$$

Strict monotonicity of $\hat{F}^k$ then implies that $x_S^{k+1} = x_M^k$. Obviously, it then holds that $\Delta_k = 0$. Also, since $x_S^{k+1} = x_M^k$ solves the subproblem $\mathrm{VI}(\hat{F}^k, S_g)$, Proposition 1 implies that this point is a solution of $\mathrm{VI}(F, S_g \cap S_h)$. □

We next establish the properties of solutions of the relaxed master problems.

Proposition 3 *For a solution (x_M^{k+1}, z^{k+1}) of the relaxed master problem $VI(F_M^k, D_k)$ in Algorithm 2, the following assertions hold:*

1. *There exists the unique Lagrange multiplier μ_M^{k+1} associated to the h-constraint. Moreover, it holds that*

$$\begin{gathered} z^{k+1} = \mu_M^{k+1}/\zeta_k, \\ 0 \leq \mu_M^{k+1} \perp h(x_M^{k+1}) - \mu_M^{k+1}/\zeta_k \leq 0. \end{gathered} \tag{24}$$

2. *The point* x_M^{k+1} *solves* $VI(F(\cdot)+[h'(\cdot)]^\top\mu_M^{k+1}, conv\, X^{k+1})$.
3. *If for any* $x \in S_g$ *and any* $v \in [F(x_M^{k+1})+[h'(x_M^{k+1})]^\top\mu_M^{k+1}] \cap [-\mathscr{N}_{conv\, X^{k+1}}(x_M^{k+1})]$ *it holds that* $\langle v, x - x_M^{k+1}\rangle < 0$, *then* $x \notin conv\, X^{k+1}$.

Proof We have that

$$0 \in F(x_M^{k+1}) \times \{\zeta_k z^{k+1}\} + \mathscr{N}_{D_k}(x_M^{k+1}, z^{k+1}).$$

Since $\operatorname{conv} X^{k+1}$ is a polyhedral set, it is easy to see that the constraints of D_k satisfy the Slater constraint qualification. Therefore,

$$\mathscr{N}_{D_k}(x_M^{k+1}, z^{k+1}) = \left\{ \begin{bmatrix} [h'(x_M^{k+1})]^\top \\ -I \end{bmatrix} \mu : \begin{array}{c} \mu \in \mathbb{R}_+^q \\ 0 \le \mu \perp h(x_M^{k+1}) - z^{k+1} \le 0 \end{array} \right\} + \left\{ \begin{pmatrix} d \\ 0 \end{pmatrix} : d \in \mathscr{N}_{\operatorname{conv} X^{k+1}}(x_M^{k+1}) \right\}.$$

In particular,

$$\zeta_k z^{k+1} - \mu = 0,$$

for any multiplier μ associated to the h-constraints. Hence, μ_M^{k+1} is uniquely defined and satisfies (24).

Also, it holds that

$$0 \in F(x_M^{k+1}) + [h'(x_M^{k+1})]^\top \mu_M^{k+1} + \mathscr{N}_{\operatorname{conv} X^{k+1}}(x_M^{k+1}),$$

which establishes the second assertion.

The last assertion follows from the same considerations as those used in Proposition 1 for Algorithm 1. □

We note that for Algorithm 2 the condition $x_S^{k+1} = x_M^k$ no longer implies that x_M^k solves $\mathrm{VI}(F, S_h \cap S_g)$ (although $\Delta_k = 0$ still implies that $x_S^{k+1} = x_M^k$ when $\hat{F}^k$ is strictly monotone). This is the price to pay for the convenience of relaxing the h-constraint, as the master problem solutions x_M^k may no longer belong to S_h. Rather, they approach this set asymptotically.

We are now in position to state the main convergence results for the Dantzig–Wolfe schemes described above, which extend [16, Thm. 8] and [5,16, Thm. 10a].

Theorem 1 *Let the mapping F be (possibly set-valued) maximal monotone or single-valued continuous, and let the function h be convex and continuously differentiable. Suppose the sequence* $\{(x_M^k, \mu_M^k, x_S^{k+1})\}$ *generated by Algorithm 1 or Algorithm 2, with the subproblems* (7) *solved approximately in the sense of* (14) *with the associated error sequence* $\{e^k\}$ *satisfying* (15) *or* (17)*, is bounded. In the case of Algorithm 2, let* $\{\zeta_k\} \to +\infty$ *as* $k \to \infty$.

Then it holds that

1. *The sequence* $\{\Delta_k\}$ *converges to zero.*
2. *If* $\hat{F}^k$ *are strongly monotone uniformly with respect to k and the approximation rule* (17) *is used, or if the matrices* Q_k *are uniformly positive definite and either* (15) *or* (17) *is used, then* $\lim_{k\to\infty} \|x_S^{k+1} - x_M^k\| = 0$.
3. *If* $\lim_{k\to\infty} \|x_S^{k+1} - x_M^k\| = 0$, *the sequence* $\{Q_k\}$ *is bounded, the families of functions* $\{F^k\}$, $\{H^k\}$ *and* $\{\mu^k\}$ *are equicontinuous on compact sets, and in the case of the relative-error inexactness rule condition* (15) *is strengthened to* (16)*, then for every accumulation point* $(\bar{x}, \bar{\mu})$ *of the sequence* $\{(x_M^k, \mu_M^k)\}$ *the point* $\bar{x}$ *is a solution of* $VI(F, S_h \cap S_g)$ *while* $\bar{\mu}$ *is a multiplier associated to the h-constraint.*

Proof Using (20) in the case when subproblems are solved inexactly according to the rule (15), and (23) if the rule (17) is employed, we see that

$$\bar{\Delta} = \liminf_{k\to\infty} \Delta_k \le \limsup_{k\to\infty} \Delta_k \le 0. \tag{25}$$

Let $\{k_j\}$ be any subsequence of indices such that $\lim_{j\to\infty} \Delta_{k_j} = \bar{\Delta}$. Passing onto a further subsequence, if necessary, we can assume that $\{(x_M^{k_j}, \mu_M^{k_j}, x_S^{k_j+1})\} \to (\bar{x}, \bar{\mu}, \hat{x})$. Also, since under the stated assumptions F is locally bounded on S_g, the sequence $\{w_M^{k_j}\}$ is bounded and we can assume that $\{w_M^{k_j}\} \to \bar{w}$.

By definition (19), we have that

$$\lim_{j\to\infty} \Delta_{k_j} = \bar{\Delta} = \langle \bar{w} + [h'(\bar{x})]^\top \bar{\mu}, \hat{x} - \bar{x} \rangle.$$

Fix any index j. Then for every $i > j$ we have that $x_S^{k_j+1} \in X^{k_i}$. As a result, by the second item of Proposition 1 in case of Algorithm 1 or of Proposition 3 in case of Algorithm 2, it holds that

$$\langle w_M^{k_i} + [h'(x_M^{k_i})]^\top \mu_M^{k_i}, x_S^{k_j+1} - x_M^{k_i} \rangle \ge 0.$$

Passing onto the limit as $i \to \infty$ in the relation above, we conclude that

$$\langle \bar{w} + [h'(\bar{x})]^\top \bar{\mu}, x_S^{k_j+1} - \bar{x} \rangle \ge 0.$$

Now passing onto the limit as $j \to \infty$ in the latter relation, we obtain that

$$\langle \bar{w} + [h'(\bar{x})]^\top \bar{\mu}, \hat{x} - \bar{x} \rangle \ge 0.$$

Hence, $\bar{\Delta} \ge 0$. Together with (25) this proves the first assertion.

Since x_S^{k+1} solves approximately $\mathrm{VI}(\hat{F}^k, S_g)$ in the sense of (14), there exists $v_S^{k+1} \in \hat{F}^k(x_S^{k+1}) \cap [-\mathcal{N}_{S_g}](x_S^{k+1})$ such that $\langle v_S^{k+1} + e^k, x_M^k - x_S^{k+1} \rangle \ge 0$. Then denoting $v_M^k = w_M^k + [h_2'(x_M^k)]^\top \mu_M^k \in \hat{F}^k(x_M^k)$ we have that

$$\begin{aligned} -\Delta_k &= \langle v_S^{k+1} - v_M^k, x_S^{k+1} - x_M^k \rangle - \langle v_S^{k+1}, x_S^{k+1} - x_M^k \rangle \\ &\ge \langle v_S^{k+1} - v_M^k, x_S^{k+1} - x_M^k \rangle + \langle e^k, x_S^{k+1} - x_M^k \rangle \\ &\ge c\|x_S^{k+1} - x_M^k\|^2 + \langle e^k, x_S^{k+1} - x_M^k \rangle, \end{aligned}$$

where $c > 0$ is the modulus of strong monotonicity of $\hat{F}^k$, independent of k. For the approximation rule (17) (that is $\{e^k\} \to 0$), since $\Delta_k \to 0$ as established above, it follows that $\|x_S^{k+1} - x_M^k\| \to 0$ as $k \to \infty$. The same conclusion holds for the choice of uniformly positive definite Q_k, as in that case the approximations $\hat{F}^k$ are uniformly strongly monotone. When the inexactness rule (15) is used, the assertion follows from (20) and the fact that $\Delta_k \to 0$.

Let $(\bar{x}, \bar{\mu})$ be an accumulation point of $\{(x_M^k, \mu_M^k)\}$ and let $\{(x_M^{k_j}, \mu_M^{k_j})\} \to (\bar{x}, \bar{\mu})$ be any associated convergent subsequence. By construction of the algorithm, the basic continuity argument implies that $\bar{x} \in S_g$ and $\bar{\mu} \in \mathbb{R}_+^q$. Since $x_S^{k_j+1}$ solves approximately $\mathrm{VI}(\hat{F}^{k_j}, S_g)$ in the sense of (14), there exists $v_S^{k_j+1} \in \hat{F}^{k_j}(x_S^{k_j+1}) \cap [-\mathcal{N}_{S_g}(x_S^{k_j+1})]$ such that

$$\langle v_S^{k_j+1} + e^{k_j}, x - x_S^{k_j+1} \rangle \ge 0 \quad \text{for all } x \in S_g. \tag{26}$$

Since the families $\{F^k\}$, $\{H^k\}$ and $\{\mu^k\}$ are equicontinuous and the matrices Q_k are bounded, the family $\{\hat{F}^k\}$ remains equicontinuous on compact sets. Then, on an open ball containing the sequences $\{x_S^{k_j+1}\}$, $\{x_M^{k_j}\}$ and the point $\bar{x}$, for every $\varepsilon > 0$ there is $\delta > 0$ such that

$\|x-y\| < \delta$ implies $d_H(\hat{F}^k(x), \hat{F}^k(y)) < \varepsilon$, for every k. Since $\|x_S^{k_j+1} - x_M^{k_j}\| \to 0$, there is an index J such that for every $j > J$ the relation $\|x_S^{k_j+1} - x_M^{k_j}\| < \delta$ holds, and thus there exists $u_M^{k_j} \in F(x_M^{k_j})$ such that $\|u_M^{k_j} + [h'(x_M^{k_j})]^\top \mu_M^{k_j} - v_S^{k_j+1}\| < \varepsilon$. On the other hand, since under the stated assumptions F is locally bounded on S_g and outer semicontinuous, we can assume that the sequence $\{u_M^{k_j}\}$ converges to a point $\bar{u} \in F(\bar{x})$. Then

$$\lim_{k\to\infty} v_S^{k_j+1} = \bar{u} + [h_2'(\bar{x})]^\top \bar{\mu}. \tag{27}$$

Next, note that $\{e^k\} \to 0$. In the case of the inexactness rule (17) this is explicit. In the case of rule (16) it is an obvious consequence since $\{Q_k\}$ is bounded, $\|x_M^k - x_S^{k+1}\| \to 0$, and the right-hand side of (16) is quadratic in the latter quantity while the left-hand side is linear.

Now passing onto the limit in (26) as $j \to \infty$ and using (27), we obtain that

$$\langle \bar{u}, x-\bar{x}\rangle + \langle [h'(\bar{x})]^\top \bar{\mu}, x-\bar{x}\rangle \geq 0 \quad \text{for all } x \in S_g. \tag{28}$$

Since $\bar{\mu} \geq 0$, the convexity of h implies that

$$\begin{aligned}\langle \bar{\mu}, h(x) - h(\bar{x})\rangle &\geq \langle \bar{\mu}, h'(\bar{x}), x-\bar{x}\rangle \\ &= \langle [h'(\bar{x})]^\top \bar{\mu}, x-\bar{x}\rangle.\end{aligned}$$

Then, by (28), we obtain that

$$\langle \bar{u}, x-\bar{x}\rangle + \langle \bar{\mu}, h(x) - h(\bar{x})\rangle \geq 0 \quad \text{for all } x \in S_g.$$

It then holds that

$$\langle \bar{u}, x-\bar{x}\rangle \geq \langle \bar{\mu}, h(\bar{x})\rangle \quad \text{for all } x \in S_h \cap S_g. \tag{29}$$

For a sequence generated by Algorithm 1, $\{x_M^k\} \subset S_h \cap S_g$ and $\langle \mu_M^k, h(x_M^k)\rangle = 0$ for all k. Hence, by continuity, $\bar{x} \in S_h \cap S_g$ and $\langle \bar{\mu}, h(\bar{x})\rangle = 0$. For a sequence generated by Algorithm 2, taking the limit in (24) as $k \to \infty$ and recalling the parameter choice $\{\zeta_k\} \to +\infty$, it again follows that $\bar{x} \in S_h \cap S_g$ and $\langle \bar{\mu}, h(\bar{x})\rangle = 0$.

In either case, we have $\bar{x} \in S_h \cap S_g$ and $\bar{u} \in F(\bar{x})$, with (29) yielding

$$\langle \bar{u}, x-\bar{x}\rangle \geq 0 \quad \text{for all } x \in S_h \cap S_g,$$

i.e., $\bar{x}$ is a solution of $\mathrm{VI}(F, S_h \cap S_g)$, as stated. The fact that $\bar{\mu}$ is a Lagrange multiplier associated to the h-constraint follows from (28). □

It is clear that the families of functions $\{F_{\text{const}}^k\}$, $\{H_{\text{const}}^k\}$ and $\{H^k{}_{\text{exact}}\}$ defined above are equicontinuous on compact sets. For a bounded sequence $\{x_M^k\}$, both $\{F_{\text{N}}^k\}$ and $\{H_{\text{N}}^k\}$ are equicontinuous. When F is single-valued continuous, the family $\{F_{\text{exact}}^k\}$ is equicontinuous. Finally, $\mu^k(x) = \mu_M^k$ is always equicontinuous while the augmented Lagrangian option $\tilde{\mu}^k(x)$ for linear constraints is equicontinuous if the sequence of penalization parameters $\{r_k\}$ is bounded.

4 Numerical Results

In this section, we describe a simplified game-theoretical model for electricity markets, and present our numerical results for computing the associated variational equilibria. For more sophisticated but related models we refer to [21–23]. For our purposes of validating the decomposition approach to variational inequalities (rather than solving the electricity models as such) the version considered here is sufficient.

4.1 Energy markets as generalized Nash games

Let N_a agents generate electric energy for sale. The i-th agent owns n_i plants whose total generation is represented by a vector $q^i \in \mathbb{R}^{n_i}_+$. The energy owned by this agent is the sum of the generation of all of the agent plants:

$$\mathsf{e}(q^i) = \sum_{k=1}^{n_i} q^i_k \,.$$

The unitary energy price in the market depends on the total amount of energy produced by all agents. We model it by a quadratic concave function of one variable, that is the total energy, so $p : \mathbb{R} \to \mathbb{R}$. The exogenous coefficients defining this quadratic function are market-dependent and are given below.

The vector of all the agents' generation is denoted by $q^{-0} = (q^1, q^2, \ldots, q^{N_a}) \in \mathbb{R}^n$ (this peculiar notation will be clear soon). The total amount of energy available in the market, denoted below by $\mathsf{e}(q^{-0})$, is the sum of the generation of all of the plants in the market:

$$\mathsf{e}(q^{-0}) = \sum_{i=1}^{N_a} \mathsf{e}(q^i) = \sum_{i=1}^{N_a} \sum_{k=1}^{n_i} q^i_k \,.$$

Since the price depends on the total energy, the the i-th agent will be paid

$$p\big(\mathsf{e}(q^{-0})\big)\mathsf{e}(q^i).$$

If, to generate the amount q^i, the agent incurs an operating (convex) cost $c^i(q^i)$, the agent's profit is given by

$$p\big(\mathsf{e}(q^{-0})\big)\mathsf{e}(q^i) - c^i(q^i).$$

The profit of each agent depends on the generation level of all the agents in the market. In turn, each generation level is constrained by technological limitations of the power plants: for certain sets $Q_i \subset \mathbb{R}^{n_i}$, the relation $q^i \in Q_i$ must hold. In our simplified modelling, $Q^i = [0, U^i]$ for some $U^i \in \mathbb{R}^{n_i}_+$, noting that in a realistic model the set Q_i is given by complex relations expressing how different technologies (thermal, nuclear, hydraulic, eolic) generate power.

Remark 1 The core difficulty for solution methods that do not use decomposition resides precisely in the fact that they handle the set $\prod_{i=1}^{N_a} Q_i$ as a whole. From a numerical point of view, this usually means dealing *simultaneously* with mixed-integer variables and non-convex relations. By contrast, a suitable decomposition method handles the difficulties by considering separately each technology (only thermal, only nuclear, etc), dealing with each set Q_i individually. As a result, an individual subproblem becomes "more computationally tractable"; for example, involving only affine functions and mixed-integer variables, or only nonlinear functions with continuous variables. Such separation of difficulties considerably simplifies the numerical solution of large problems. In a somewhat different context, this is also confirmed by our results below.

An additional constraint for the generation levels q^i refers to the fact that agents are encouraged to satisfy the market demand $d > 0$. We let $q^0 \geq 0$ denote a scalar slack variable, measuring the deficit of energy in the market, sometimes called *load shedding*. Then, if for each agent i, the vector

$$q^{-i} = (q^0, q^1, \ldots, q^{i-1}, q^{i+1}, \ldots, q^{N_a})$$

denotes the generation level of all the other agents, including load shedding, the relation

$$q^i \in D(q^{-i}) = Q^i \cap \left\{ w^i \in \mathbb{R}^{n_i} : q^0 + \sum_{i \neq j=1}^{N_a} \sum_{k=1}^{n_j} q_k^j + \sum_{k=1}^{n_i} w_k^i = d \right\}$$

must hold.

Summing up, the i-th agent tries to maximize profit by solving the (concave) problem

$$\max \quad p\big(\mathtt{e}(q^{-0})\big)\mathtt{e}(q^i) - c^i(q^i) \quad \text{s.t.} \quad q^i \in D(q^{-i}).$$

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 (this is a so-called *bounded rationality* model). Accordingly, letting the ISO be player number 0, if the energy deficit is penalized with a price $P > 0$, the ISO tries to maximize the social welfare by solving

$$\max \quad p\big(\mathtt{e}(q^{-0})\big)\mathtt{e}(q^{-0}) - \sum_{i=1}^{N_a} c^i(q^i) - Pq^0 \quad \text{s.t.} \quad q^0 \in D(q^{-0}),$$

where, having a maximal allowed level of load shedding U^0,

$$D(q^{-0}) = \left\{ w^0 \in \mathbb{R} : 0 \leq w^0 \leq U^0, \quad w^0 + \mathtt{e}(q^{-0}) = w^0 + \sum_{i=1}^{N_a} \sum_{k=1}^{n_i} q_k^i = d \right\}.$$

As a result, for $i = 0, \ldots, N_a$, the convex functions $\theta_i : \mathbb{R}^{1+n} \to \mathbb{R}$ given by

$$\begin{aligned} \theta_0(q) &= Pq^0 - \sum_{i=1}^{N_a} c^i(q^i) - p\big(\mathtt{e}(q^{-0})\big)\mathtt{e}(q^{-0}) \\ \theta_i(q) &= c^i(q^i) - p\big(\mathtt{e}(q^{-0})\big)\mathtt{e}(q^i), \qquad\qquad i = 1, \ldots, N_a, \end{aligned}$$

define a generalized Nash game with $N_a + 1$ players (the ISO and the N_a agents). In this game, each player tries to maximize profit by solving

$$\min \quad \theta_i(q^{-i}, q^i) \quad \text{s.t.} \quad q^i \in D(q^{-i}).$$

It is known that finding a generalized Nash equilibrium (GNE) of this game is equivalent to solving a quasi-variational inequality problem, see [12]. Quasi-variational problems are very hard to solve. Fortunately, in our case, it is possible to compute some GNE points (not all) by solving a variational inequality instead. These points are called *variational equilibria* and have some good/important properties from the economic point of view, see [13, 25]. For our problem, it is shown in [26] that variational equilibria are solutions to $\mathrm{VI}(F, S_g \cap S_h)$ where

$$F(q) = \Big(\nabla_{q^0} \theta_0(q), \nabla_{q^1} \theta_1(q), \ldots, \nabla_{q^{N_a}} \theta_{N_a}(q) \Big),$$

$$S_g = [0, U^0] \times \prod_{i=1}^{N_a} Q_i \quad \text{and} \quad S_h = \{ q \in \mathbb{R}^{1+n} : q^0 + \mathtt{e}(q^{-0}) = d \}.$$

In our model each function $\theta_i(q)$ is convex and differentiable in the variable q^i. So the function $\nabla_{q^i} \theta_i(q)$ is monotone in the i-th component of the variable q, but in general it is not monotone on the full variable q. Therefore, the singled-valued function F defining the variational problem (1) is nonmonotone. Observe that it also couples all the variables.

Our Dantzig-Wolfe strategy can be applied to nonmonotone single-valued functions F, simply by taking monotone approximations to F in the subproblems. In particular, any family F^k_{const}, $F^k_{\text{N-J}}$, or $F^k_{\text{exact-J}}$ can be used.

Another specificity of our game is that the demand constraint, that is the set S_h, couples all variables q^i. Without this constraint, the feasible set would be separable. This makes the considered game particularly suitable for application of our decomposition schemes.

Remark 2 As discussed above, the model considered here is simplistic in some features; it is mostly meant to exhibit the interest of using decomposition schemes for problems with feasible sets having certain types of structure. In particular, the shared constraint in our model refers to satisfaction of the (exogenous) demand, but alternative joint constraints, like the ones in [18], could also be considered.

4.2 Battery of problems

We implemented Algorithm 1 in Matlab version 7.11(R2010b). The runs were done on a PC operating under Ubuntu 11.04 with a Core(TM)2 Duo 2.00GHz processor and 4GB of memory.

We created six market configurations of the generalized Nash game, by taking $N_a = 5$ agents and considering a mix with n power plants, for

$$n \in \{100, 250, 1000, 2500, 5000, 10000\}.$$

As n increases, the configurations become harder and harder to solve directly, without decomposition. Also the subproblems become harder, as we assume that each agent owns the same number of plants $n_i = n/5$, for $i = 1, \ldots, N_a = 5$.

Other values of the model parameters are as follows.

1. The entries of the maximum generation capacity vector U^i are random numbers in $[0, 10]$ while the maximum allowed deficit is fixed to $U^0 = 5$.
2. The demand is taken equal to $d = 0.8 \sum_{i=1}^{N_a} U^i$, corresponding to 80% of the market generation capacity.
3. The deficit price is set at $P = 120$.
4. The unitary price p is a quadratic concave function such that $p(0) = P$, $p'(0) = 0$ and $p(1.5d) = 0$.
5. The operating cost is of the form

$$c^i(q^i) = b^{i\top} q^i + \frac{1}{2} q^{i\top} M_i q^i$$

where $b^i \in \mathbb{R}^{n_i}$ and $M_i \in \mathbb{R}^{n_i \times n_i}$ is a diagonal positive definite matrix. The corresponding values are generated randomly between $[30, 60]$ and $[0.4, 0.8]$, respectively.

With this data, the simplified model is set up in a manner ensuring that at a variational equilibrium of the game will have no deficit ($q^0 = 0$) and the price will be equal to $p(d) = P(1 - \frac{1}{1.5^2})$. As a way of ensuring correctness of the implementation, we checked that these values were obtained in all of our numerical results below.

4.3 Results

For each of the six market configurations, we randomly generated 10 problem instances. For each instance, we apply Algorithm 1 using five approximations for F:

$$F_{\text{const}}\,, \quad F_{\text{N}}\,, \quad F_{\text{exact}} = F\,, \quad F_{\text{N-J}}\,, \quad F_{\text{J}}\,.$$

Since the h-constraint is linear, we used the exact family H_{exact} and likewise for the multipliers.

We also tried to solve the problem directly, without decomposition, using PATH [9,15]. For the two largest configurations ($n = 5000, 1000$) PATH could no longer be used, stopping by lack of memory. With our computer and for the considered instances, when $n = 5000$ the solver stalled after about 4 hours. Also, since the larger configurations become time consuming, for these we only run the faster decomposition alternatives, in our case F_{const}, F_{J} and $F_{\text{N-J}}$.

Regarding specifics of the implementation of Algorithm 1, all the subproblems and master problems are themselves solved using PATH. Our focus here is on comparing various approximation options; for this reason we do not report on the variants with inexact solution of subproblems, with generating additional points via projections, or relaxing the master problems.

For the cases $n = 100, 250, 500, 1000$, and the options F_{const}, $F_{\text{N-J}}$, and $F_{\text{exact-J}}$, subproblems in variables q^i followed the decomposition pattern induced by the product $S_g = \prod_{i=0}^{5} S_{g_i}$, where $S_{g_0} = [0, U^0] \subset \mathbb{R}$ and $S_{g_i} = Q_i \subset \mathbb{R}^{n/5}$, $i \geq 1$. With this decomposition, decision variables are precisely those of each player. For the larger configurations we used instead the product $S_g = \prod_{i=0}^{n/250} S_{g_i} = [0, U^0] \times \prod_{i=1}^{5} Q_i$, with $S_{g_0} = [0, U^0] \subset \mathbb{R}$ and $S_{g_i} \subset \mathbb{R}^{250}$.

We use as stopping rules the following criteria. In PATH the stopping test employs the residual of the full problem based on the Fischer-Burmeister merit function [14, Chapter 1.5] with a default 10^{-6}. For the decomposition approaches the stopping criterion is

$$\frac{|\Delta_k|}{1 + |\Delta_1|} < 10^{-5},$$

where Δ_k is defined in (19). As discussed in Section 2.4, this is a natural stopping condition in the decomposition framework, as the access to the full problem, and thus to its residual, is not presumed.

All results are reported in Table 2, and interpreted in the two comparative Figures 1 and 2 below. For each configuration, we averaged over the 10 instances the results for each method. The table reports the average and maximal CPU times in seconds; the percentage of the total running time spent in the master and subproblem solution; the mean residual (the infinite-norm of the natural merit function [14, Chapter 1.5] for $\text{VI}(F(\cdot) + [h'(\cdot)]^\top \mu_M^k, S_g)$ at the master solution x_M^k); and the mean infinite-norm of the difference between x_M^k and x_S^{k+1} at termination. In particular, the latter distance and the residual are not a part of the decomposition stopping test (as they are not available within the decomposition scheme anyway); these values were computed *a posteriori*, to confirm that an approximate solution of the problem was indeed obtained.

Regarding the running times, the main point we would like to stress is that for the configurations with $n \geq 5000$, applying PATH directly appears no longer possible even after relaxing the stopping tolerance from the default 10^{-6} to 10^{-2}. Of course, the exact threshold depends on the specific computer and implementation, yet there is always a threshold.

Size and Model	CPU_{MEAN} (s)	CPU_{MAX} (s)	Master (% time)	SubPbm (% time)	Residual	$\|x_S - x_M\|_\infty$
$n = 100$						
PATH	0.162	0.235	-	-	-	-
F_{const}	5.199	8.081	92	8	0.001	3.006
F_{N}	1.170	1.331	72	27	0.016	0.048
F	1.973	2.276	46	53	0.015	0.051
$F_{\text{N-J}}$	2.010	4.923	89	10	0.083	0.069
F_{J}	8.785	10.027	17	83	0.080	0.068
$n = 250$						
PATH	0.303	0.326	-	-	-	-
F_{const}	9.002	12.617	94	6	0.007	6.114
F_{N}	4.397	4.819	23	77	0.016	0.043
F	2.530	2.691	41	59	0.015	0.045
$F_{\text{N-J}}$	1.915	2.274	80	20	0.070	0.125
F_{J}	9.561	11.107	16	84	0.075	0.123
$n = 500$						
PATH	1.413	1.713	-	-	-	-
F_{const}	19.413	30.356	95	5	0.008	8.071
F_{N}	29.079	31.620	6	93	0.017	0.035
F	7.229	8.359	25	75	0.017	0.036
$F_{\text{N-J}}$	3.313	4.067	64	36	0.087	0.099
F_{J}	14.285	20.401	15	85	0.086	0.100
$n = 1000$						
PATH	13.807	13.962	-	-	-	-
F_{const}	54.202	66.884	97	3	0.009	7.641
F_{N}	236.751	250.975	2	98	0.015	0.029
F	47.541	49.154	9	91	0.015	0.030
$F_{\text{N-J}}$	11.263	13.878	40	60	0.067	0.074
F_{J}	40.398	48.539	11	89	0.066	0.073
$n = 2500$						
PATH	693.439	695.626	-	-	-	-
F_{const}	255.843	275.134	98	2	0.025	8.918
F_{N}	4590.224	4919.049	0	100	0.022	0.037
F	923.633	1069.350	2	98	0.022	0.038
$F_{\text{N-J}}$	37.028	42.773	53	47	0.058	0.101
F_{J}	335.984	383.007	6	94	0.057	0.085
PATH	-	-	-	-	-	-
F_{const}	1043.489	1257.464	99	1	0.038	9.535
$F_{\text{N-J}}$	114.774	123.597	83	17	0.032	0.056
F_{J}	2239.637	2408.383	4	96	0.031	0.055
PATH	-	-	-	-	-	-
F_{const}	4204.447	4450.205	99	1	0.066	9.857
$F_{\text{N-J}}$	483.478	575.581	72	28	0.029	0.046
F_{J}	13891.376	14891.010	2	98	0.029	0.045

Table 2 Detailed list of all results

On the other hand, some of the approximation options in the decomposition technique still succeed in solving the larger configurations in reasonable computational times.

The column reporting maximal CPU times in Table 2 gives an estimation on how the data dispersion affected each method. For $n = 2500$, for example, the percentual difference with respect to the mean CPU time was of 0.3, 7.5, 7.2, 15.8, 15.5, and 14.0%, respectively for PATH and the constant, Newton, Newton-Jacobi, and Jacobi approximations. As expected, the impact of varying data on the constant approximation is much smaller than for the other

approximations, which incorporate more information. The situation is similar for $n = 10000$, where the percentages are 5.8, 19.1, and 7.2 for the only three approaches that could solve such large instances, respectively F_{const}, $F_{\text{N-J}}$, and F_{J}.

In order to see the benefit of decomposing, we took as reference the CPU time taken by a direct application of PATH and computed the ratio between the CPU times of each decomposition method and the reference one. Figure 1 shows the corresponding ratios. We should remark though that Figure 1 is intended merely to illustrate the dynamics of the comparison as the size grows and should not be taken literally. The reason is that, being a Newton-type method, when PATH works (i.e., for problems not too large), it provides highly accurate solutions. Generally, a comparable level of accuracy cannot be expected from the decomposition approach.

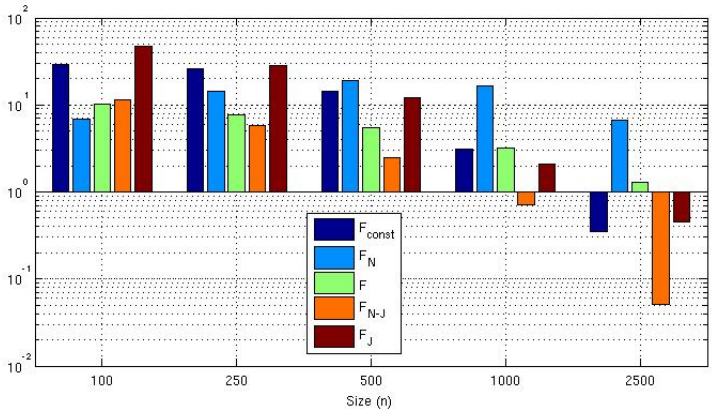

Fig. 1 Time ratios (decomposition divided by PATH), configurations with $n \leq 2500$

The ordinate in Figure 1 uses a logarithmic scale, for convenience. In the figure, when for a given method the value represented by a color bar lies above the 0-ordinate (corresponding to 10^0 in the logarithmic scale), the decomposition method took longer than the direct approach with PATH. By contrast, when the bar is below the ordinate 10^0, the decomposition method was faster than PATH. The plot in Figure 1 shows a natural behaviour. A direct application of PATH is very efficient for the smaller to medium sized problems. But as the size grows, decomposition becomes more and more competitive. For $n = 1000$, the $F_{\text{N-J}}$ decomposition already outperforms the direct approach. And for $n = 2500$, three of the decomposition approaches become faster than PATH, with the $F_{\text{N-J}}$ approximation being the best one.

When the percentual distribution of time between master and subproblem solution in Table 2 does not add up to 100%, this is due to some time spent in intermediate tasks, such as communicating with PATH mex-interface. We observe that for the larger configurations in general the best approach (Newton-Jacobi) spends less time in solving subproblems than in dealing with the master problems. In view of our comments in Remark 1, we conjecture that if we were to consider difficult sets Q_i, the percentual distribution of time would result in higher figures for the subproblems. Since the Jacobi-like approximations are amenable to parallelization (thus making the subproblem solution quicker), for such decompositions the more CPU time is spent in solving subproblems in our current serial implementation, the

faster would be the overall procedure in parallel implementation. Moreover, it is also likely that more intrincate sets Q_i would make decomposition preferable over a direct solution with PATH even for the smaller instances (always keeping in mind that this is a problem dependent issue).

Regarding solution quality, the two last columns in Table 2 report, respectively, the value of the *a posteriori* computed residual and of the gap between x_S and x_M. We observe that while the constant approximation gives systematically the most distant x_S and x_M's, for smaller instances this approximation also has the lowest residual. This tendency starts changing at $n = 2500$ and for $n \geq 5000$ the Newton-Jacobi and Jacobi approximations become more accurate, and practically equally so. Since the approximation F_{N-J} is the fastest one, it appears as the best option for large configurations, both in terms of speed and accuracy.

We finish this section with an analysis of scalability. Recall once again that PATH applied to the full problem stalls for $n \geq 5000$, while some decomposition approaches still work in reasonable time. Figure 2 compares the performance of F_{const}, $F_{\text{N-J}}$, F_{J}, which are the options able to handle the larger configurations. The plot shows the corresponding mean CPU times in minutes for each configuration, ranging from $n = 100$ to $n = 10000$. We observe that $F_{\text{N-J}}$ shows the best scalability with respect to the problem size, suggesting once more this is the best option for larger models.

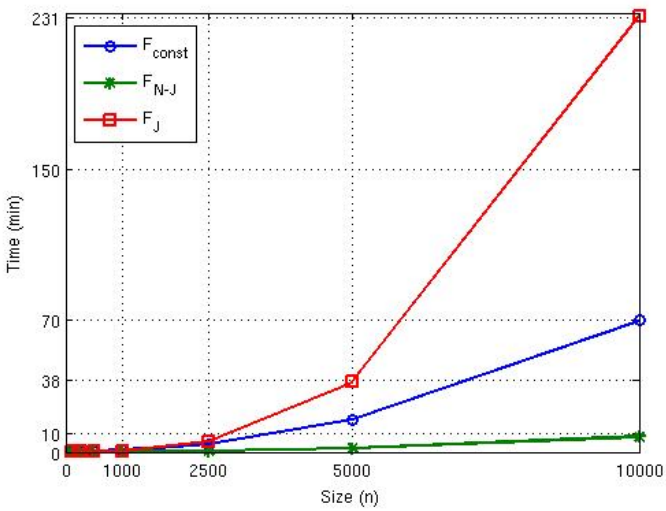

Fig. 2 Scalability of the best decomposition options

Concluding remarks

We have presented a family of decomposition methods for variational inequalities, that can be applied with maximal monotone operators (possibly set-valued) or single-valued continuous operators (possibly nonmonotone). The approach allows for various kinds of approxi-

mations of the problem data and its derivatives in the single-valued case, as well as inexact solution of subproblems. The resulting algorithmic patterns are shown to be convergent under reasonable assumptions on the variational problem.

The decomposition scheme is highly versatile and makes it possible to exploit structural properties of the variational problem, even if the operator therein is not separable. The benefit of having such a flexible setting is assessed by our numerical experiments, computing a variational equilibrium for generalized Nash games arising when modeling strategic interactions in electricity markets.

Our numerical results show that, even with an extremely simplified modelling of the power system, large instances become untractable with a direct solution method, and can only be solved by decomposition. In terms of accuracy, speed, and scalability, among all the considered variants and for our battery of tests, the decomposition method using a combination of Newton and Jacobi approximations appears to be the best one.

Acknowledgements We thank the two referees for their constructive comments which helped us to improve the original version.

References

1. H. Ben Amor, J. Desrosiers, and A. Frangioni. On the Choice of Explicit Stabilizing Terms in Column Generation. *Discrete Applied Mathematics*, 157:1167–1184, 2009.
2. J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. Numerical Optimization: Theoretical and Practical Aspects. Springer-Verlag, Berlin, Germany, 2006 (2nd edition).
3. O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113:299–344, 2008.
4. G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathematical Programming*, 64:81–101, 1994.
5. W. Chung and J.D. Fuller. Subproblem approximation in Dantzig-Wolfe decomposition of variational inequality models with an application to a multicommodity economic equilibrium model. *Operations Research*, 58:1318–1327, 2010.
6. R.W. Cottle, J.-S. Pang, and R.E. Stone. The Linear Complementarity Problem. Academic Press, Inc., Boston, MA, 1992.
7. T.G. Crainic, A. Frangioni, and B. Gendron. Bundle-based Relaxation Methods for Multi-commodity Capacitated Fixed Charge Network Design Problems. *Discrete Applied Mathematics*, 112:73–99, 2001.
8. G.B. Dantzig and P. Wolfe. The decomposition algorithm for linear programs. *Econometric*, 29:767–778, 1961.
9. S.P. Dirkse and M.C. Ferris. The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems. *Optimization Methods and Software*, 5:123–156, 1995.
10. J. Eckstein and D.P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.
11. J. Eckstein. Some saddle-function splitting methods for convex programming. *Optimization Methods and Software*, 4:75–83, 1994.
12. F. Facchinei and C. Kanzow. Generalized Nash equilibrium problems. *Ann. Oper. Res.* 175:177–211, 2010.
13. F. Facchinei, A. Fischer and V. Piccialli. On generalized Nash games and variational inequalities. *Oper. Res. Lett.*, 35:159–164, 2007.
14. F. Facchinei, J.S. Pang. Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer–Verlag, New York, NY, 2003.
15. M.C. Ferris and T.S. Munson. Interfaces to PATH 3.0: Design, implementation and usage. *Computational Optimization and Applications*, 12:207–227, 1999.
16. J.D. Fuller and W. Chung. Dantzig–Wolfe Decomposition of Variational inequalities. *Computational Economics*, 25:303–326, 2005.
17. B. He, L.Z. Liao, D. Han and H. Yang. A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming*, 92:103–118, 2002.
18. B. F. Hobbs and J. S. Pang. Nash-Cournot Equilibria in Electric Power Markets with Piecewise Linear Demand Functions and Joint Constraints *Operations Research*, 55(1):113–127, 2007.

19. K.L. Jones, I.J. Lustig, J.M. Farwolden, and W.B. Powell. Multicommodity Network Flows: The Impact of Formulation on Decomposition. *Mathematical Programming*, 62:95–117, 1993.
20. N.H. Josephy. Newton's method for generalized equations. *Technical Summary Report no. 1965. Mathematics Research Center, University of Wisconsin, Madison*, 1979.
21. A. Kannan and V.M. Zavala. A Game-Theoretical dynamic model for electricity markets. Argonne National aboratory, Technical Report ANL/MCS-P1792-1010, 2011.
22. A. Kannan, U.V. Shanbhag, and H.M. Kim. Addressing supply-side risk in uncertain power markets: stochastic generalized Nash models, scalable algorithms and error analysis. *Optimization Methods and Software*, to appear.
23. A. Kannan, U.V. Shanbhag, and H.M. Kim. Strategic Behavior in Power Markets under Uncertainty. *Energy Systems*, 2: 115–141, 2011.
24. I.V. Konnov. Combined Relaxation Methods for Variational Inequalities (Lecture Notes in Economics and Mathematical Systems, Vol.495). Springer-Verlag, Berlin-Heiderberg-New York, 2001.
25. A. A. Kulkarni, U. V. Shanbhag On the variational equilibrium as a refinement of the generalized Nash equilibrium. *Automatica*, 48: 45-55, 2012.
26. A. A. Kulkarni, U. V. Shanbhag Revisiting generalized Nash games and variational inequalities. *Journal of Optimization Theory and Applications*, 2012. DOI 10.1007/s10957-011-9981-5.
27. P.A. Lotito, L.A. Parente and M.V. Solodov. A class of variable metric decomposition methods for monotone variational inclusions. *Journal of Convex Analysis*, 16:857–880, 2009.
28. Z.-Q. Luo and P. Tseng. Error bound and convergence analysis of matrix splitting algorithm for the affine variational inequality problems. *SIAM Journal on Optimization*, 2:43–54, 1992.
29. B. Martinet. Regularisation d'inequations variationelles par approximations successives. *Revue Française d'Informatique et de Recherche Opérationelle*, 4:154–159, 1970.
30. L.A. Parente, P.A. Lotito and M.V. Solodov. A class of inexact variable metric proximal point algorithms. *SIAM Journal on Optimization*, 19:240–260, 2008.
31. T. Pennanen. A splitting method for composite mappings. *Numerical Functional Analysis and Optimization*, 23:875–890, 2002.
32. R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976.
33. R.T. Rockafellar. Local boundedness of nonlinear, monotone operators. *Michigan Math. J.*, 16:397–407, 1969.
34. R.T. Rockafellar. On the maximality of sums of nonlinear monotone operators. *Trans. Amer. Math. Soc.*, 149:75–88, 1970.
35. U.V. Shanbhag, G. Infanger, and P.W. Glynn. A complementarity framework for forward contracting under uncertainty. *Operations Research*, 59: 810–834, 2011.
36. M.V. Solodov. A class of globally convergent algorithms for pseudomonotone variational inequalities. In *Complementarity: Applications, Algorithms and Extensions*, M C. Ferris, O.L. Mangasarian and J.-S. Pang (editors), Applied Optimization 50, Kluwer Academic Publishers, 2001, Chapter 14, pp. 297–315.
37. M.V. Solodov. A class of decomposition methods for convex optimization and monotone variational inclusions via the hybrid inexact proximal point framework. *Optimization Methods and Software*, 19:557–575, 2004.
38. M.V. Solodov. Constraint qualifications. In *Wiley Encyclopedia of Operations Research and Management Science*, James J. Cochran, et al. (editors), John Wiley & Sons, Inc., 2010.
39. M.V. Solodov and B.F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical Functional Analysis and Optimization*, 22:1013–1035, 2001.
40. M.V. Solodov and P. Tseng. Modified projection-type methods for monotone variational inequalities. *SIAM Journal on Control and Optimization*, 34:1814–1830, 1996.
41. J.E. Spingarn. Applications of the method of partial inverses to convex programming. *Mathematical Programming*, 32:199–223, 1985.
42. P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29:119–138, 1991.
43. P. Tseng. Alternating projection-proximal methods for convex programming and variational inequalities. *SIAM Journal on Optimization*, 7:951–965, 1997.
44. P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38:431–446, 2000.