Solutions to Affine Generalized Equations Using Proximal Mappings *

Stephen C. Billups†and Michael C. Ferris ‡

November, 1994 (Revised May, 1996)

Abstract

The normal map has proven to be a powerful tool for solving generalized equations of the form: find $z \in C$, with $0 \in F(z) + N_C(z)$, where C is a convex set and $N_C(z)$ is the normal cone to C at z. In this paper, we use the T-map, a generalization of the normal map, to solve equations of the more general form: find $z \in \text{dom}(T)$, with $0 \in F(z) + T(z)$, where T is a maximal monotone multifunction. We present a path-following algorithm that determines zeros of coherently oriented piecewise-affine functions, and we use this algorithm, together with the T-map, to solve the generalized equation for affine, coherently oriented functions F, and polyhedral multifunctions T.

1 Introduction

This paper is concerned with solving generalized equations [20, 21] of the form

$$0 \in F(x) + T(x) \tag{1}$$

where T is a maximal monotone multifunction from $\mathbb{R}^n$ into $\mathbb{R}^n$ and F is a continuously differentiable function from an open set $\Omega \supset \text{dom}(T)$ into $\mathbb{R}^n$. We recall that a monotone multifunction T is a point to set mapping such that for each $(x^1, y^1), (x^2, y^2)$ in the graph of T,

$$\left\langle x^1 - x^2, y^1 - y^2 \right\rangle \geq 0,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, T is maximal if its graph is not properly contained in that of any other monotone multifunction and $\text{dom}(T)$ represents the effective domain of T.

To date, most of the algorithmic development for generalized equations has been focused on the special case where $T := N_C$, the normal cone to a convex set C, defined by

$$T(z) = N_C(z) := \begin{cases} \{w \mid \langle w, y - z \rangle \leq 0, \quad \forall y \in C\} & z \in C \\ \emptyset & z \notin C \end{cases}$$

This case yields the generalized equation

$$0 \in F(x) + N_C(x). \tag{2}$$

*This material is based on research supported by National Science Foundation Grant CCR-9157632, the Air Force Office of Scientific Research Grant F49620-94-1-0036, and the Department of Energy Grant DE-FG03-94ER61915

†Mathematics Department, University of Colorado, Denver, Colorado 80217

‡Computer Sciences Department, University of Wisconsin, Madison, Wisconsin 53706

Many problems from mathematical programming, equilibrium, complementarity and other fields can be expressed in this form. For example, if $F := \nabla f$, then (2) represents the first order necessary optimality conditions for the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C. \end{array}$$

Another important instance of (2) is the variational inequality problem, which is to find $z \in C$ such that

$$\langle y - z, F(z) \rangle \geq 0, \qquad \forall y \in C.$$

This problem is known to be equivalent to (2) (see [21]).

As a final example, we mention the complementarity problem; this has a wealth of applications and appears in a variety of forms [13]. The standard nonlinear complementarity problem is to find $x \in \mathrm{R}^n_+$ such that

$$F(x) \geq 0, \quad \text{and} \quad \langle x, F(x) \rangle = 0.$$

It is well known [13, 15] that complementarity problems can be reformulated as variational inequalities, and therefore can be treated in the context of the generalized equation (2).

A prominent tool for solving (2) is the *normal map* [8, 22, 26]. The normal map F_C for a function $F : \Omega \to \mathrm{R}^n$ and a nonempty, closed, convex set $C \supset \Omega$ is defined by

$$F_C(x) := F(\pi_C(x)) + x - \pi_C(x), \tag{3}$$

where $\pi_C(x)$ denotes the projection (with respect to the Euclidean norm) of x onto the set C. The importance of the normal map lies in the fact that solving (2) is equivalent to finding a zero of the normal map F_C. Specifically, if x is a zero of the normal map, then $z := \pi_C(x)$ solves (2). Conversely, if z is a solution to (2), then $x := z - F(z)$ is a zero of the normal map. Thus, the problem of solving the generalized equation (2), which is expressed with set-valued functions, is reduced to finding a zero of a piecewise smooth, single-valued function.

Several algorithms have been developed based on this idea. Most of these algorithms are based on the theory of piecewise linear homotopies given by Eaves [9]. The specialization of the general algorithm given in [9] determines a zero of the normal map by tracing the zero curve of a piecewise linear homotopy mapping. It is shown in [12] that Lemke's famous method [16] for solving linear complementarity problems is conceptually equivalent to path following in a corresponding piecewise linear system. This idea is easily extended to more general algorithms [10, 11]. Based on Eaves' work, Cao and Ferris [3] analyzed an algorithm for solving affine (i.e. $F(x)$ is affine) variational inequality problems over polyhedral sets and showed that Lemke's method processed matrices in the class P_C [2].

For non-affine problems with C rectangular, Ralph [18] proposed a Newton-based algorithm where at each iteration the Newton point is calculated by solving an affine variational inequality (AVI) that is a linearization of the normal map equation at the current point. This approach was developed computationally by Dirkse and Ferris [5, 6, 7] to produce PATH, an efficient and robust code for solving mixed complementarity problems. A similar approach was developed by Rutherford [24].

In this paper, we begin to generalize the above class of algorithms by removing the restriction that T be the normal cone to a convex set. As a first step, in this paper we focus on solving the *affine generalized equation* where F is an affine function and T is polyhedral (that is, the graph of T is the union of finitely many convex polyhedrons).

Our strategy is, in essence, a generalization of the AVI algorithm of Cao and Ferris. We use a generalization of the normal map to reformulate (1) as a zero finding problem of a piecewise affine function; then we use the piecewise-linear homotopy framework of Eaves [9] to generate an algorithm for determining zeros of piecewise-affine functions. We describe this algorithm in detail in Section 2 and prove that under the assumption of coherent orientation, the algorithm finds a zero after a finite number of steps.

The remainder of the paper describes how the algorithm is used to solve affine generalized equations. In Section 3 we discuss a generalization of the normal map called the T-map. This mapping, which to our knowledge was introduced by Minty [17], provides a means of reformulating generalized equations involving operators T that do not necessarily correspond to the normal cone of any set. After describing this reformulation, we focus on the case where T is polyhedral. This case can be viewed as an extension of the special case where $T := N_C$, with C a convex polyhedron. In Section 4 we describe how the algorithm presented in Section 2 can be used to solve affine generalized equations. In Section 5 we further specialize to the case where T is separable. Finally, in Section 6 we discuss how the algorithm can be applied to solve piecewise linear-quadratic programming problems [25].

Some words about notation are needed. The notation $T : \mathbf{R}^n \rightrightarrows \mathbf{R}^n$ indicates that T is a point to set mapping, or *multifunction*, which maps points in $\mathbf{R}^n$ to subsets of $\mathbf{R}^n$. In discussing multifunctions, we may refer either to a point to set mapping, $T(\cdot)$, or to the graph of that mapping, T, which is the set $T := \{(x, y) \mid y \in T(x)\}$. The expression $T^{-1}(\cdot)$ is defined as a set inverse; i.e., $T^{-1}(y) := \{x | (x, y) \in T\}$. Further, $T^{-1} = \{(y, x) | (x, y) \in T\}$. The effective domain of T, is defined by $\operatorname{dom}(T) := \{x | T(x) \neq \emptyset\}$.

In discussing matrices and vectors, subscripts are used to refer to components. For example $A_{i\cdot}, A_{\cdot j}, A_{ij}$ refer to the ith row, jth column, and (i, j)th entry of A, respectively. We may also use index sets to refer to specific components. For example if $\alpha = \{1, 3, 5\}$, then $x_\alpha = \{x_1; x_3; x_5\}$. Further, we use the MATLAB notation of a comma to separate columns, and a semicolon to separate rows. For example (x, y) is a row vector, whereas $(x; y)$ is a column vector. Unless otherwise indicated, all vectors are taken to be column vectors. Superscripts are used to indicate an iteration count, index, or some other label for matrices and vectors. In contrast, for scalars, sets, and functions, subscripts are used as labels.

For a set C, $\operatorname{aff}(C)$, $\operatorname{int}(C)$, $\operatorname{ri}(C)$, $\operatorname{rec}(C)$, $\dim(C)$, and $\delta(\cdot|C)$ refer to the affine hull, interior, relative interior, recession cone, dimension, and indicator function of C respectively. (see [23] for definitions of these terms).

2 Algorithm for Finding Zeros of $\mathcal{M}$-PA Maps

Our first task is to describe an algorithm for finding zeros of piecewise affine functions. The algorithm we present can be viewed as a generalization of Lemke's method for the linear complementarity problem and of the algorithm for solving affine variational inequalities over convex polyhedral sets that was described in [3]. The theoretical basis for the algorithm is derived from the theory of piecewise-linear homotopies given in [9].

In order to describe the algorithm carefully, we need some preliminary definitions.

Definition 2.1 (cell) *A polyhedral convex set $\sigma \subset \mathbf{R}^n$ is called a* cell. *If $\dim(\sigma) = k$ then σ is called a k-*cell. *Let $\sigma := \{x | Ax \leq a\}$, where $A \in \mathbf{R}^{p \times n}$, and $a \in \mathbf{R}^p$, with p a nonnegative integer. Then (p, A, a) is said to represent σ. If p is the smallest number for which a representation of σ exists, then (p, A, a) is called a* minimal representation *of σ. A set $\tau \in \mathbf{R}^n$ is called a* face *of σ if* *for some set of indices* $\alpha \subset \{1, \ldots, p\}$, $\tau = \{x \in \sigma : A_{\alpha}.x = b_{\alpha}\}$. *If* $\dim(\tau) = i$, *then* τ *is called an* i*-face of* σ.

Clearly any cell has a minimal representation.

Definition 2.2 (piecewise affine) *Let* $\mathcal{M}$ *be a collection of* n*-cells and let* $M := \bigcup_{\sigma \in \mathcal{M}} \sigma$. *A function* $F : M \to \mathbf{R}^m$ *is said to be* piecewise-affine *with respect to* $\mathcal{M}$, *denoted* $\mathcal{M}$-PA, *if for each* $\sigma \in \mathcal{M}$, $F_{|\sigma}$ *(i.e. the restriction of* F *to* σ*) is affine. If* F *is* $\mathcal{M}$*-PA for some* $\mathcal{M}$ *satisfying the above assumptions, then we say that* F *is piecewise affine.*

Note that in the above definition, if M is convex, then the function F must be continuous on M, because F must be single-valued on the boundaries between cells. Furthermore, in contrast to the work of Eaves [9], $\mathcal{M}$ is not required to correspond to a subdivided manifold.

Definition 2.3 (function representation) *Let* $\mathcal{M}$ *be a collection of* n*-cells in* $\mathbf{R}^n$, *let* F *be a* $\mathcal{M}$*-PA function, and let* σ *be an* n*-cell of* $\mathcal{M}$. *Let* $b^{\sigma} \in \mathbf{R}^m$ *and let* B^{σ} *be an* $m \times n$ *matrix.* (B^{σ}, b^{σ}) *is said to* represent F *on* σ *if* $F(x) = B^{\sigma}x + b^{\sigma}$ *for all* $x \in \sigma$.

We now describe an algorithm to find a zero of an $\mathcal{M}$-PA function G, for a given collection of cells $\mathcal{M}$. We will assume that representations of the cells of $\mathcal{M}$ and of the map G have already been constructed. The basis of the algorithm is to construct a piecewise affine homotopy mapping $F(x, \mu)$ with the following properties

1. $(x^*, 0)$ is a zero of F if and only if x^* is a zero of G.
2. A point (x^1, μ_1), and a direction $(d^1, -1)$ is known such that $\mu_1 \geq 0$ and $F(x^1 - \mu d^1, \mu^1 + \mu) = 0$ for all $\mu \geq 0$.

The algorithm uses a method described by Eaves [9] to trace the zero curve of F, proceeding in the direction $(d^1, -1)$ from the starting point (x^1, μ_1). To prove that the algorithm finds a solution in a finite number of steps, we restrict ourselves to the case where G is *coherently oriented*:

Definition 2.4 (coherent orientation) *Let* G *be an* $\mathcal{M}$*-PA map with representation* (B^{σ}, b^{σ}) *on each* $\sigma \in \mathcal{M}$. *We say that* G *is* coherently oriented *if*

$$\operatorname{sgn}(\det(B^{\sigma}))$$

is nonzero and constant for all σ *in* $\mathcal{M}$, *where*

$$\operatorname{sgn}(x) := \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0. \end{cases}$$

Since $\mathcal{M}$ is finite and $\bigcup_{\sigma \in \mathcal{M}} \sigma = \mathbf{R}^n$, it follows that $\mathbf{R}^n = \bigcup_{\sigma \in \mathcal{M}} \operatorname{rec}(\sigma)$, and further that there is a σ such that $\operatorname{int}(\operatorname{rec}(\sigma)) \neq \emptyset$. Choose d such that $-d \in \operatorname{int}(\operatorname{rec}(\sigma))$. Then for any x^0 in $\mathbf{R}^n$, and for all μ sufficiently large, $x^0 - \mu d \in \operatorname{int}(\sigma)$.

In the AVI algorithm described by Cao and Ferris, the cell σ and the direction d were constructed by finding an extreme point x^e of the set C. The cell was then given by $\sigma := x^e + N_C(x^e)$, and the direction d was chosen such that $-d$ was in the interior of $N_C(x^e)$. For our algorithm, rather than constructing the cell and direction, we can rely instead on the fact that since $\mathbf{R}^n = \bigcup_{\sigma \in \mathcal{M}} \operatorname{rec}(\sigma)$, then for any direction d, there will be a cell σ_d for which $-d \in \text{rec}(\sigma_d)$. Note further that for each cell σ, the boundary $\text{rec}(\sigma) \setminus \text{int}(\text{rec}(\sigma))$ of $\text{rec}(\sigma)$ has Lebesgue measure zero. Therefore, since the number of cells is finite, $\bigcup_{\sigma \in \mathcal{M}} \text{rec}(\sigma) \setminus \text{int}(\text{rec}(\sigma))$ has measure zero. Thus, for almost all d, there will be a cell σ_d for which $-d \in \text{int}(\text{rec}(\sigma_d))$.

Thus, if x^0 is any point in R^n, then for all μ sufficiently large, $x^0 - \mu d$ will lie interior to the cell σ_d. In other words, the cell can be chosen simply by picking an arbitrary d and proceeding in the direction $-d$ until a cell σ_d is reached for which $-d$ is in the recession cone of σ_d. For almost all d (excepting a set of Lebesgue measure zero), $-d$ will be in the *interior* of $\text{rec}(\sigma_d)$. We note, however, that for some special cases, construction of an extreme point may still be preferable.

Once d and σ_d have been identified, the homotopy map can be constructed. Let (B, b) be the representation of G in σ_d. Define a function $F : \mathrm{R}^n \times \mathrm{R}_+ \to \mathrm{R}^n$ by

$$F(x, \mu) := G(x) + \mu B d. \tag{4}$$

Note that $F(x, 0) = 0$ exactly when $G(x) = 0$. Under the assumption that G is coherently oriented, B is invertible. Let $x^0 := -B^{-1}b$ and define

$$w(\mu) = x^0 - \mu d.$$

Then, since $-d \in \text{int}(\text{rec}(\sigma_d)))$, there exists $\mu_0 \geq 0$ such that $w(\mu) \in \text{int}(\sigma_d), \forall \mu > \mu_0$. Thus, for $\mu \geq \mu_0$,

$$\begin{aligned} F(w(\mu), \mu) &= G(w(\mu)) + \mu B d \\ &= B w(\mu) + b + \mu B d \\ &= B(x^0 - \mu d) + b + \mu B d \\ &= -b - \mu B d + b + \mu B d \\ &= 0. \end{aligned} \tag{5}$$

By choosing $\mu_1 > \mu_0$, $x^1 = w(\mu_1)$, and $d^1 = d$, we see that F satisfies the conditions needed for the homotopy map. We are now ready to state the algorithm, which is given in Figure 1. Note that by normalizing d in the discussion above to be a unit vector, we can start the algorithm from the point (x^1, μ_1) constructed above with $\sigma_1 := \sigma_d$.

Some comments about Algorithm AGE are in order:

1. Most of the work in the algorithm is in step 8 where the direction (d^{k+1}, v_{k+1}) is calculated. At the end of this section, in Theorem 2.13, we show that $B^{k+1} - B^k$ is a rank-1 matrix. Thus, an efficient implementation of the algorithm can be obtained by keeping the matrices B^k in factored form and performing rank-1 updates of the factors at each step of the algorithm.

2. At step 8 in the algorithm, there may be more than one possible choice of cells σ_{k+1}. However, a lexicographic ordering, as described by Eaves [9, Section 15], can be used to resolve any ambiguity concerning which cell to choose. The use of such a lexicographic ordering will be assumed in the convergence proof, and will be presented in more detail in the discussion preceding Lemma 2.9.

3. The requirement that $\left\| d^{k+1} \right\| = 1$ is arbitrarily chosen to force the choice of d^{k+1} to be unique.

4. The requirement that $x^1 - \mu d^1 \in \text{int}(\sigma_1), \forall \mu \geq 0$ guarantees that the zero curve of $F(x, \mu) := G(x) + \mu B d^1$ contains a ray, and therefore assures us that it will not have any loops. This fact will be useful in our convergence proof. However, we shall also show that, under the assumption of coherent orientation, v_k is always negative, which by itself guarantees that no

Figure 1: Algorithm AGE

Given a finite collection of n-cells $\mathcal{M}$ such that $\bigcup_{\sigma \in \mathcal{M}} \sigma = \mathbf{R}^n$, and an $\mathcal{M}$-PA function G on R^n. Let G have representation (B^k, b^k) on $\sigma_k \in \mathcal{M}$.

1) **[Initialization]** Determine $(x^1, \mu_1, \sigma_1, d^1)$ satisfying
   $\|d^1\| = 1$,
   $x^1 \in \mathrm{int}(\sigma_1)$,
   $x^1 - \mu d^1 \in \mathrm{int}(\sigma_1), \quad \forall \mu \geq 0$,
   $B^1 x^1 + \mu_1 B^1 d^1 + b^1 = 0$.
2) $v_1 := -1$.

Repeat for $k = 1, 2, \ldots$

3) $\theta_k := \sup\{\theta | x^k + \theta d^k \in \sigma_k, \mu_k + \theta v_k \geq 0\}$.
   If $\theta_k = +\infty$, then
   4) **output**("ray termination"); **return**.
   Else
   5) $x^{k+1} := x^k + \theta_k d^k$
   6) $\mu_{k+1} := \mu_k + \theta_k v_k$
      If $\mu_{k+1} = 0$ then
      7) **output**("solution found at", x^{k+1}); **return**.
      Else
      8) determine σ_{k+1} (possibly using lexicographic ordering),
         d^{k+1}, and v_{k+1} such that
            $x^{k+1} \in \sigma_{k+1}$,
            $B^{k+1} d^{k+1} + v_{k+1} B^1 d^1 = 0$,
            $\left\| d^{k+1} \right\| = 1$,
            d^{k+1} points into σ_{k+1} from x^{k+1},
            and $\sigma_{k+1} \in \mathcal{M} \setminus \sigma_k$.
      9) goto next k.

---

loops occur. Thus, under the assumption of coherent orientation, it is not necessary to find a ray start. However, in future work, we will prove convergence for a broader class of problems, in which case the ray start requirement will be useful.

The next few pages are devoted to proving the following convergence theorem:

Theorem 2.5 *Let $\mathcal{M}$ be a finite collection of n-cells whose relative interiors are disjoint and whose union is R^n. Let G be a coherently oriented, $\mathcal{M}$-PA function. Algorithm AGE, using lexicographic ordering, terminates after finitely many steps with a zero x^* of G.*

Proof (Outline) There are three main parts to the proof. First, as Lemma 2.9, we will show that the algorithm terminates at a solution if $\mathcal{M}$ is a *subdivision* of R^n (see Definition 2.6). Second, in Lemma 2.11, we will show that even if $\mathcal{M}$ is not a subdivision of R^n, there is a *refinement* (see Definition 2.10) $\mathcal{N}$ of $\mathcal{M}$ that is a subdivision. Finally, we show in Lemma 2.12 that if a subdivision $\mathcal{N}$ is a refinement of $\mathcal{M}$, then running the algorithm using $\mathcal{N}$ will generate exactly the same path as would be generated by using $\mathcal{M}$. Thus, the fact that the algorithm terminates at a solution using $\mathcal{N}$ guarantees that it will terminate at a solution using $\mathcal{M}$. $\square$

We now prove the three lemmas mentioned above. At this point, we recommend that the impatient reader skip ahead to Theorem 2.13. Our proof technique is based on the work of Eaves [9]. Eaves' analysis relies heavily on the notion of a subdivided manifold:

Definition 2.6 (subdivided manifold) *Let N be a set in some Euclidean space, and let $\mathcal{N}$ be a finite or countable collection of n-cells in that space such that $N = \bigcup_{\sigma \in \mathcal{N}} \sigma$. Let $\tilde{\mathcal{N}}$ be the collection of all faces of elements of $\mathcal{N}$. $(N, \tilde{\mathcal{N}})$ is a* subdivided n-manifold *if*

1. *any two n-cells of $\mathcal{N}$ are either disjoint or meet in a common face;*
2. *each point of N has a neighborhood meeting only finitely many n-cells of $\mathcal{N}$;*
3. *each $(n-1)$-cell of $\tilde{\mathcal{N}}$ lies in at most two n-cells;*

If $(N, \tilde{\mathcal{N}})$ is a subdivided n-manifold for some subdivision $\mathcal{N}$, we call N an n-manifold *and we call $\mathcal{N}$ a* subdivision *of N.*

The following lemma shows that when $N = \mathrm{R}^n$, item 3 in Definition 2.6 is redundant. This result was proved by Robinson [22] in the proof of Proposition 2.4. While Robinson's proposition is stated for the normal manifold, his proof is valid for general subdivisions of R^n.

Lemma 2.7 *If $\mathcal{N}$ is a collection of cells whose union is R^n and if $\mathcal{N}$ satisfies 1 and 2 of Definition 2.6, then $\mathcal{N}$ is a subdivision of R^n.*

The next step in our analysis is to prove that the algorithm works whenever $\mathcal{M}$ is a subdivision of R^n. In this case, by defining $\mathcal{S} := \{\sigma \times \mathrm{R}_+ | \sigma \in \mathcal{M}\}$, we see that $\mathcal{S}$ is a subdivision of $\mathrm{R}^n \times \mathrm{R}_+$ and further that F is $\mathcal{S}$-PA. The starting point (x^1, μ_1) of the algorithm lies interior to the cell $\eta_1 := \sigma_1 \times \mathrm{R}_+$ of $\mathcal{S}$. Further, the ray $\{(x^1, \mu_1) - \mu(d^1, -1) | \mu \geq 0\}$ lies within η_1. Let $\tilde{\mathcal{S}}$ be the collection of all faces of elements of $\mathcal{S}$. Algorithm AGE is then seen to be equivalent to the algorithm described by Eaves [9, Section 10.2], with the following relationships between the algorithms:

Eaves' Algorithm	Algorithm AGE
$\mathcal{M}$	$\tilde{\mathcal{S}}$
$F(x)$	$F(x,\mu) := G(x) + \mu B^1 d^1$
x_k	(x^k, μ_k)
σ_k	η_k
v_k	(d^k, v_k)

To discuss the behavior of this algorithm in more detail, we need some definitions from [9].

Definition 2.8 (regularity) *Let $(N, \tilde{\mathcal{N}})$ be a subdivided $(n+1)$-manifold, let $\mathcal{N}$ be the collection of n-cells in $\tilde{\mathcal{N}}$, and let $F : N \to \mathrm{R}^n$ be a $\mathcal{N}$-PA map. A point x in N is said to be* degenerate *(otherwise* regular*) if x lies in a cell σ of $\tilde{\mathcal{N}}$ with $\dim(F(\sigma)) < n$. A value y in $F(N)$ is said to be a* degenerate value *(otherwise a* regular value*) if $F^{-1}(y)$ contains a degenerate point.*

Note that if y is a regular value, then $F^{-1}(y)$ cannot intersect any k-cells of $\mathcal{S}$ with $k < n$.

By the assumption of coherent orientation, G is one-to-one in every n-cell of $\mathcal{M}$. Thus, $\dim(F(\eta)) = n$ for all $(n+1)$-cells η of $\mathcal{S}$. Since the starting point (x^1, μ_1) of the algorithm is interior to η_1, it is a regular point of F. According to [9, Theorem 15.13], since $\mathcal{S}$ is finite, the algorithm generates, in finitely many steps, either a point (x^*, μ_*) in the boundary of $\mathbf{R}^n \times \mathrm{R}_+$, or a ray in $F^{-1}(0)$ different from the starting ray. In the first case, we know that $\mu_* = 0$, since the boundary of $\mathrm{R}^n \times \mathrm{R}_+$ is $\mathrm{R}^n \times \{0\}$. It then follows, from our earlier remarks that x^* satisfies $G(x^*) = 0$. Therefore, to guarantee that the algorithm finds a solution, we need only show that it cannot produce a ray different from the starting ray.

We first consider the case when 0 is a *regular value* of F. In this case, by [9, Theorem 9.1], $F^{-1}(0)$ is a 1-manifold which is subdivided by sets of the form $\eta \bigcap F^{-1}(0)$. Further, since $F^{-1}(0)$ cannot intersect any k-cells with $k < n$, each point on $F^{-1}(0)$ is in at most two $(n+1)$-cells of $\mathcal{S}$. Thus, in step 8 of the algorithm, the choice of σ_{k+1} is well-defined. (The only difficulty would be if (x^{k+1}, μ_{k+1}) lies in only one $(n+1)$-cell η_k so that no σ_{k+1} could be selected. But in this case, (x^{k+1}, μ_{k+1}) would be a boundary point of $\mathrm{R}^n \times \mathrm{R}_+$. Thus, $\mu_{k+1} = 0$, so the algorithm would have terminated in step 5.)

Let (d^k, v_k) be the direction of the path within the $(n+1)$-cell η_k of $\mathcal{S}$, and let G have representation (B^k, b^k) on the n-cell σ_k of $\mathcal{M}$. Then by [9, Lemma 12.3], the *curve index*, given by

$$(\mathrm{sgn} v_k)(\mathrm{sgn}\det B^k)$$

is constant everywhere along the path. Since $v_1 = -1$ for the starting direction (d^1, v_1), and since G is coherently oriented, it follows that v_k is negative in each cell that the path enters. But this means that the parameter μ decreases strictly in each cell. Thus, after finitely many steps, we must have $\mu = 0$.

When 0 is a degenerate value of F, $F^{-1}(0)$ may intersect a k-cell of $\mathcal{S}$ with $k < n$. Thus, in step 8 of the algorithm, there may be multiple choices for which cell σ_{k+1} to enter next. To address this problem, a lexicographic ordering can be used to resolve ambiguities concerning which cell the path will enter. Such a scheme is conceptually equivalent to solving a perturbed problem, which we now describe.

Let $X = [\xi^1, \ldots, \xi^n]$ be an $(n+1) \times n$ matrix such that $[X, d^1]$ is of rank $(n+1)$. Define the vector $[\epsilon] := (\epsilon^1, \epsilon^2, \ldots, \epsilon^n)^\top$ (note: the superscripts here refer to exponentiation). Define $(x^1(\epsilon); \mu_1(\epsilon)) := (x^1; \mu_1) + X[\epsilon]$. Since $(x^1; \mu_1)$ in Algorithm AGE is interior to η_1, then $(x^1(\epsilon); \mu_1(\epsilon)) \in \mathrm{int}(\eta_1)$ for small enough ϵ. Further, since $(-d^1, 1) \in \mathrm{int}(\mathrm{rec}(\eta_1))$, $(x^1(\epsilon), \mu_1(\epsilon)) + \mu(-d^1, 1) \in \mathrm{int}(\eta_1)$, for all $\mu \geq 0$. Thus, $x^1(\epsilon), \mu_1(\epsilon), \sigma_1$, and d^1 satisfy the starting conditions needed to apply the algorithm to the perturbed problem given by

$$0 = F(x, \mu) - p(\epsilon),$$

where $p(\epsilon) := F(x^1(\epsilon), \mu_1(\epsilon))$. Observe that

$$\begin{aligned} p(\epsilon) &= F(x^1, \mu_1) + [B^1, B^1 d^1] X[\epsilon] \\ &= [B^1, B^1 d^1] X[\epsilon] \\ &= Y[\epsilon], \end{aligned} \tag{6}$$

where $Y := [B^1, B^1 d^1] X$. Y is an invertible $n \times n$ matrix, so that by [9, Lemma 14.2], $p(\epsilon)$ is a regular value for all ϵ sufficiently small. Thus, by the arguments given above for regular values, using Algorithm AGE to solve the perturbed problem will, after a finite number of steps J, produce a point $(x^J(\epsilon))$ such that $G(x^J(\epsilon)) = p(\epsilon)$.

Let $(x^k(\epsilon), \mu_k(\epsilon))$ be the sequence of points generated by the algorithm for the perturbed problem. By the discussion in [9, Section 15], there is a sequence of matrices $X^k \in \mathrm{R}^{(n+1)\times n}$ and a sequence of points (x^k, μ_k) such that $(x^k(\epsilon); \mu_k(\epsilon)) := (x^k; \mu_k) + X^k[\epsilon]$ for all small ϵ. The points (x^k, μ_k) are exactly the sequence of points generated by the algorithm for solving the unperturbed problem using the lexicographic ordering. Since the algorithm terminates after J steps for all small ϵ, we see that $\mu_J(\epsilon) = 0$ and $G(x^J(\epsilon)) = p(\epsilon)$. It follows that $\mu_J = 0$ and further that $G(x^J) = 0$. Thus, using a lexicographic ordering, the algorithm finds a solution after a finite number of steps.

We have proved the following lemma:

Lemma 2.9 *Let $\mathcal{M}$ be a subdivision of R^n and A_T be a coherently oriented, $\mathcal{M}$-PA function. Algorithm AGE, using lexicographic ordering, terminates after finitely many steps with a zero x^* of G.*

We now address the case where $\mathcal{M}$ is not a subdivision of R^n. We begin by proving that $\mathcal{M}$ can be *refined* to produce a subdivision.

Definition 2.10 (refinement) *Let $\mathcal{M}$ and $\mathcal{N}$ be finite collections of n-cells. $\mathcal{N}$ is said to be a* refinement *of $\mathcal{M}$ if each cell σ of $\mathcal{M}$ is the union of a finite collection of cells τ_i of $\mathcal{N}$, and if each cell of $\mathcal{N}$ is contained in some cell of $\mathcal{M}$.*

The following lemma is proved by Hudson [14, Lemma 1.5], however, using different nomenclature. In particular, the term "subdivision" is used in place of our term "refinement".

Lemma 2.11 *Let $\mathcal{M} := \{C_i\}$ be a collection of $J < \infty$ n-cells which covers R^n. There exists a subdivision $\mathcal{N}$ of R^n such that $\mathcal{N}$ is a refinement of $\mathcal{M}$.*

We now show that using $\mathcal{N}$, the algorithm follows the same path as it would using $\mathcal{M}$.

Lemma 2.12 *Let G be an $\mathcal{M}$-PA function, where $\mathcal{M}$ is a finite collection of relatively disjoint n-cells whose union is R^n. Let $\mathcal{N}$ be a refinement of $\mathcal{M}$ such that $\mathcal{N}$ is a subdivision of R^n. Then Algorithm AGE, using lexicographic ordering, will find a solution x^* to $G(x) = 0$ in a finite number of steps. Furthermore, the sequence of points generated by the algorithm using $\mathcal{M}$ is a subsequence of the points that would be generated using $\mathcal{N}$.*

Proof Consider first running the algorithm using $\mathcal{N}$ instead of $\mathcal{M}$. By Lemma 2.9, the algorithm will terminate after some finite number of steps J. The algorithm will visit a sequence of n-cells $\{\tau_k\} \subset \mathcal{N}$, and will generate a sequence of points $\{(x^k, \mu_k)\}$ and directions $\{(d^k, v_k)\}$, for $k = 1, \ldots, J$.

Let $j_1 := 1$ and let σ_1 be the unique cell in $\mathcal{M}$ that contains τ_1. Then for $i = 2, \ldots$, let j_i be the smallest index greater than j_{i-1} such that $\tau_{j_i} \not\subset \sigma_{i-1}$, and if $j_i \le J$, let σ_i be the unique cell in $\mathcal{M}$ that contains τ_{j_i}. Let K be such that $j_K = J$. This process defines a sequence of cells $\{\sigma_i\}, i = 1, \ldots, K$ and indices $\{j_i\}, i = 1, \ldots, K+1$ such that $\tau_k \subset \sigma_i$ whenever $j_i \le k < j_{i+1}$.

We will show that if the algorithm is run using $\mathcal{M}$, then the sequence of points $\{(\xi^i, \nu_i)\}$ generated by the algorithm satisfies the equation $(\xi^i, \nu_i) = (x^{j_i}, \mu_{j_i})$, for each i. Thus, $(\xi^K, \nu_K) = (x^J, \mu_J)$, so the algorithm finds a solution after a finite number of steps.

Let $\{(\delta^i, \zeta_i)\}$ be the sequence of directions chosen by the algorithm using $\mathcal{M}$. Clearly, since the algorithm is started at the point (x^1, μ_1) in the direction $(d^1, -1)$, the following is true: $(\xi^1, \nu_1) = (x^{j_1}, \nu_{j_1})$, $(\delta^1, \zeta_1) = (d^{j_1}, v_{j_1})$, and the first cell visited by the algorithm is σ_1.

We now proceed by induction: Assume that $(\xi^i, \nu_i) = (x^{j_i}, \nu_{j_i})$, $(\delta^i, \zeta_i) = (d^{j_i}, v_{j_i})$, and that, using $\mathcal{M}$, the ith cell visited by the algorithm is σ_i. We shall prove that $(\xi^{i+1}, \nu_{i+1}) = (x^{j_{i+1}}, \nu_{j_{i+1}})$, $(\delta^{i+1}, \zeta_{i+1}) = (d^{j_{i+1}}, v_{j_{i+1}})$, and that the $(i+1)$st cell visited by the algorithm is σ_{i+1}.

Let (B^i, b^i) be the representation of G on σ_i. This is also the representation of G on τ_k whenever $j_i \le k < j_{i+1}$. Thus, in step 8 of the algorithm using $\mathcal{N}$, the direction (d^{k+1}, v_{k+1}) chosen when entering cell τ_{k+1} must satisfy

$$B^i d^{k+1} + v_{k+1} B^1 d^1 = 0$$

for $j_i \le k < j_{i+1}$. Since G is coherently oriented, B^i is invertible. Further, $\left\|d^{k+1}\right\| = 1$ and, by our earlier discussion, v_{k+1} is negative. Thus, the direction is uniquely determined by the representation. In particular, $(d^{j_i}, v_{j_i}) = (d^{j_i+1}, v_{j_i+1}) = \cdots = (d^{j_{i+1}}, v_{j_{i+1}}) = (\delta^i, \zeta_i)$. From this it is clear that $x^{j_{i+1}}$ lies on the ray $\{\xi^i(\theta) | \theta \ge 0\}$, where $\xi^i(\theta) := \xi^i + \theta\delta^i$. Further, $x^{j_{i+1}}$ is on the boundary of σ_i.

If the ray $\{\xi^i(\theta) | \theta \ge 0\}$ contains a point in the interior of σ_i, then the ray cannot be extended past $x^{j_{i+1}}$ without exiting σ_i. Thus, $x^{j_{i+1}} = \xi^i(\theta_i)$ where $\theta_i := \sup\{\theta | \xi^i(\theta) \in \sigma_i\}$. In other words, $(x^{j_{i+1}}, \mu_{j_{i+1}}) = (\xi^{i+1}, \nu_{i+1})$.

If the ray $\{\xi^i(\theta) | \theta \ge 0\}$ does not contain an interior point of σ_i, then we must resort to the lexicographic ordering to prove that $x^{j_{i+1}} = \xi^i(\theta_i)$. Since σ_i and $\tau_{j_{i+1}}$ are relatively disjoint convex sets, there exists a separating hyperplane H_i defined by a vector c^i, and a scalar α_i such that ${c^i}^\top x < \alpha_i, \forall x \in \mathrm{int}(\sigma_i)$, and ${c^i}^\top x \ge \alpha_i, \forall x \in \tau_{j_{i+1}}$. Suppose we run the algorithm using $\mathcal{N}$ to solve the perturbed problem $G_\epsilon(x) := G(x) - p(\epsilon) = 0$, where $p(\epsilon)$ is defined by (6). Then, for ϵ small enough, the algorithm will visit the same sequence of cells $\{\tau_k\}$ as it visits in the unperturbed problem. Also, by our earlier discussion, the algorithm will generate the sequence of points $\{(x^k(\epsilon); \mu_k(\epsilon))\} = \{(x^k; \mu_k) + X^k[\epsilon]\}$, where $\{X^k\}$ is a fixed sequence of matrices.

Since 0 is a regular value of G_ϵ, $\dim(G_\epsilon(\tau_k \bigcap \tau_{k+1})) \ge n-1$ for any k. Thus, $G_\epsilon^{-1}(0)$ contains only one point in $\tau_k \bigcap \tau_{k+1}$, namely x^{k+1}. Therefore, the direction d^{k+1} must point into the interior of τ_{k+1}.

By similar arguments as before, $x^{j_{i+1}}$ lies on the ray $\{\xi^i_\epsilon(\theta) | \theta \ge 0\}$, where $\xi^i_\epsilon(\theta) := x^{j_i}(\epsilon) + \theta(d^{j_i})$. But, since d^{k+1} points into the interior of τ_{j_i}, this ray must contain a point $\hat{x}$ in the interior of σ_i. Thus, ${c^i}^\top \hat{x} < \alpha_i$. But ${c^i}^\top x^{j_{i+1}} \ge \alpha_i$ since $x^{j_{i+1}}$ is in $\tau_{j_{i+1}}$. It follows that ${c^i}^\top d^{j_i} > 0$. Thus, even for the unperturbed problem, the ray $\xi^i(\theta)$ cannot be extended past the point $x^{j_{i+1}}$ without crossing the hyperplane H_i, and thereby exiting σ_i. Thus, $x^{j_{i+1}} = \xi^i(\theta_i)$, and as before, $(x^{j_{i+1}}, \mu_{j_{i+1}}) = (\xi^{i+1}, \nu_{i+1})$.

Finally, note that for all small ϵ, the point $x^{j_{i+1}}(\epsilon)$ is a regular point, so $\tau_{(j_{i+1}-1)}$ and $\tau_{j_{i+1}}$ are the only n-cells of $\mathcal{N}$ that contain $x^{j_{i+1}}(\epsilon)$. Thus, σ_i and σ_{i+1} are the only n-cells of $\mathcal{M}$ that contain $x^{j_{i+1}}(\epsilon)$. Thus, for all small ϵ, the algorithm, using $\mathcal{M}$ will enter cell σ_{i+1} at the next iteration. But this means that using lexicographic ordering the algorithm will enter cell σ_{i+1} next when solving the unperturbed problem. Finally, since the representation of G on σ_{i+1} is identical to the representation of G on $\tau_{j_{i+1}}$, we must have $(\delta^{i+1}, \zeta_{i+1}) = (d^{j_{i+1}}, v^{j_{i+1}})$.

The lemma is now proved by induction. $\square$

This completes the proof of Theorem 2.5. Our final task in this section is to establish the claim made in Comment 1 following Algorithm AGE.

Theorem 2.13 *Under the hypothesis of Theorem 2.5, let $\{\sigma_k\}$ be the sequence of cells chosen in Step 8 of Algorithm AGE using lexicographic ordering, and let (B^k, b^k) represent A_T on σ_k. Then $B^{k+1} - B^k$ has rank 1.*

Proof Using lexicographical ordering, the algorithm will choose the same cell σ_{k+1} in step 8 as it would when solving the perturbed problem for small ϵ. However, 0 is a regular value for the perturbed problem, so $\tau_k := \sigma_k \bigcap \sigma_{k+1}$ must have dimension $n-1$. Now, for any two points $x^1, x^2 \in \tau_k$

$$B^{k+1}(x^1 - x^2) = B^k(x^1 - x^2) \Longrightarrow (x^1 - x^2) \in \ker(B^{k+1} - B^k)$$

Thus, $\dim \ker(B^{k+1} - B^k) = n-1$ so $\operatorname{rank}(B^{k+1} - B^k) = 1$. $\square$

3 The T-map

The T-map, denoted F_T, is a generalization of the normal map that is formed by replacing the projection operator π_C in (3) by the resolvent operator $P_T := (I+T)^{-1}$. Specifically, the T-map is given by

$$F_T(x) := F(P_T(x)) + x - P_T(x). \tag{7}$$

We assume throughout that T is a maximal monotone operator. In this case, Minty [17] showed that P_T is a continuous, single-valued, nonexpansive function defined on all of R^n. Since the image of P_T is $\mathrm{dom}(T)$ (which is contained in the domain of F, Ω), it follows that F_T is a single-valued function defined on all of R^n.

By [1, Example 2.1.2], $I+T$ is monotone, and therefore P_T is monotone. We now show that solving GE(F,T) is equivalent to finding a zero of F_T.

Theorem 3.1 *Given a maximal monotone multifunction $T : \mathrm{R}^n \rightrightarrows \mathrm{R}^n$, and a function $F : \Omega \subset \mathrm{R}^n \to \mathrm{R}^n$, let F_T be defined by (7). If x is a zero of F_T, then $z := P_T(x)$ solves GE(F,T). Conversely, if z solves GE(F,T), then $x := z - F(z)$ is a zero of F_T.*

Proof Suppose $F_T(x) = 0$ and let $z := P_T(x)$. Then $0 = F_T(x) = F(z) + x - z$, and

$$\begin{aligned} -F(z) &= x - z \\ &\in (I+T)(I+T)^{-1}(x) - z \\ &= (I+T)(z) - z \\ &= T(z). \end{aligned}$$

Conversely, suppose $-F(z) \in T(z)$ and let $x := z - F(z)$. Then $x \in z + T(z) = (I+T)(z)$, so $P_T(x) = (I+T)^{-1}(x) = z$. Thus $F_T(x) = F(z) + x - z = F(z) - F(z) = 0$. $\square$

So far, we have not made any assumptions on T other than that it is maximal monotone. We now focus on the case where T is polyhedral.

Definition 3.2 (polyhedral) *A multifunction T is* polyhedral *if its graph is the union of finitely many polyhedral convex sets.*

Our first task will be to show that, for polyhedral T, the resolvent operator $P_T := (I+T)^{-1}$ is a piecewise-affine map.

Lemma 3.3 *A single-valued multifunction $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ whose graph is a convex polyhedron is affine on* $\operatorname{dom}(T)$*.*

Proof Assume $\operatorname{dom}(T) \neq \emptyset$. (Otherwise the lemma is true vacuously). Since the graph of T is a polyhedron, T can be written as $T = \{(x,y)|Ax + By \geq c\}$ for some $A \in \mathbb{R}^{p\times n}, B \in \mathbb{R}^{p\times m}$, and $c \in \mathbb{R}^p$, where p is some nonnegative integer. Let $\mathcal{K} := \{i|A_{i\cdot}x + B_{i\cdot}y = c_i, \forall(x,y) \in T\}$. In words, $\mathcal{K}$ is the set of row indices for which the corresponding constraint is active for *all* points in T.

We first establish the fact that $\ker B_{\mathcal{K}\cdot} = \{0\}$. To do this, let $\mathcal{H} := \{i|i \notin \mathcal{K}\}$. Then for each $i \in \mathcal{H}, \exists(x^i, y^i) \in T$ with $A_{i\cdot}x^i + B_{i\cdot}y^i > c_i$. Let $(\tilde{x},\tilde{y}) = \sum_{i\in\mathcal{H}}(x^i,y^i)/|\mathcal{H}|$, where $|\mathcal{H}|$ is the cardinality of the index set $\mathcal{H}$. Note that $(\tilde{x},\tilde{y})$ is a convex combination of points in T and is therefore also in T. Further, $A_{i\cdot}\tilde{x} + B_{i\cdot}\tilde{y} > c_i, \forall i \in \mathcal{H}$. Now, if $\hat{y} \in \ker B_{\mathcal{K}\cdot}$, then for $\epsilon > 0$ small enough, $A\tilde{x} + B(\tilde{y} + \epsilon\hat{y}) \geq c$. Thus, $(\tilde{x}, \tilde{y} + \epsilon\hat{y}) \in T$. But since T is single-valued, $\hat{y} = 0$. Thus, $\ker B_{\mathcal{K}\cdot} = \{0\}$.

Now, by the definition of $\mathcal{K}$, we have $(x,y) \in T \Rightarrow A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}}$. Conversely, suppose (x,y) satisfies $A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}}$. If $x \in \operatorname{dom}(T)$, then $\exists\hat{y}$ such that $(x,\hat{y}) \in T$. But this means that $A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}\hat{y} = c_{\mathcal{K}}$, which implies that $\hat{y} - y \in \ker B_{\mathcal{K}\cdot} = \{0\}$. That is $\hat{y} = y$. We have thus shown that

$$(x,y) \in T \Leftrightarrow x \in \operatorname{dom}(T) \text{ and } A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}}.$$

Finally, since $\ker B_{\mathcal{K}\cdot} = \{0\}$, $B_{\mathcal{K}\cdot}$ has a left inverse $R \in \mathbb{R}^{m\times p}$. Thus, for $x \in \operatorname{dom}(T)$,

$$\begin{aligned}(x,y) \in T \quad &\Leftrightarrow \quad A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}} \\ &\Leftrightarrow \quad y = Rc_{\mathcal{K}} - RA_{\mathcal{K}\cdot}x.\end{aligned}$$

So T is an affine function on $\operatorname{dom}(T)$. □

Theorem 3.4 *Given a maximal monotone polyhedral multifunction $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, the resolvent operator $P_T := (I+T)^{-1}$ is a piecewise affine function on all of $\mathbb{R}^n$.*

Proof Since T is polyhedral, $I+T$ is also polyhedral [19] and therefore so is $P_T = (I+T)^{-1}$. Thus, $P_T = \bigcup \Gamma_i$, where $\{\Gamma_i\}$ is a finite collection of polyhedral convex sets. Let C_i be the projection of Γ_i onto the domain of P_T (i.e., $C_i = \pi_1(\Gamma_i)$, where $\pi_1 := (x,y) \mapsto x$). Define $\mathcal{M} := \{C_i|\dim(C_i) = n\}$.

Since P_T is defined on all of $\mathbb{R}^n$, $\bigcup C_i = \mathbb{R}^n$. Let $M := \bigcup_{C_i\in\mathcal{M}} C_i$. Since M is closed, its complement, $\backslash M := \mathbb{R}^n \setminus M$, is open. Thus, $\backslash M$ is either the empty set, or it has nonempty interior. But $\backslash M \subset \bigcup_{\dim(C_i)<n} C_i$. Thus, $\backslash M$ has no interior. In other words $\backslash M = \emptyset$ and thus, $M := \mathbb{R}^n$.

To show that P_T is $\mathcal{M}$-PA, all that is needed is to show that for each $C_i \in \mathcal{M}$, the restriction of P_T to C_i is affine. However, since P_T is single-valued, the graph of P_T restricted to C_i is simply the convex polyhedral set Γ_i. By Lemma 3.3, P_T is affine on C_i. □

Corollary 3.5 *If T is polyhedral and F is affine, then the T-map, F_T, defined by (7) is piecewise affine.*

4 Affine Generalized Equations

We now show how to apply the algorithm of Section 2 to construct an algorithm to solve the *affine generalized equation*:

$$0 \in Ax - a + T(x), \tag{8}$$

where $A \in \mathbf{R}^{n\times n}$, $a \in \mathbf{R}^n$, and T is a maximal monotone polyhedral multifunction. For this problem, the T-map is given by

$$A_T := AP_T(x) + x - P_T(x) - a. \tag{9}$$

As was shown in Section 3, for polyhedral T, A_T is piecewise affine with respect to some *finite* collection $\mathcal{M}$ of n-cells whose union is R^n. Thus, to complete the description of the algorithm for affine generalized equations, it remains to show how to generate the representations.

The task of constructing $\mathcal{M}$ is dependent upon how T is described. For example, in [22], T is taken as the normal cone N_C to a polyhedral convex set C. $\mathcal{M}$ is then chosen to be the *normal manifold*, which is defined in terms of the nonempty faces F_i of C. Specifically, the cells of the normal manifold are defined by

$$\sigma_i := F_i + N_{F_i},$$

where N_{F_i} is the common value of N_{F_i} for $x \in \mathrm{ri}(F_i)$. This particular choice of cells leads to the algorithm given in [3].

For more general T, we assume that T is described as the union of a finite collection of polyhedral convex sets C_i. We can then describe P_T as the union of the sets $S_i := \{(x+y, x) \mid (x,y) \in C_i\}$. By projecting each S_i onto the domain of P_T, we produce a collection of sets

$$\sigma_i := \{x + y \mid (x,y) \in C_i\}.$$

Since we know $\mathrm{dom}(P_T) = \mathrm{R}^n$, it follows that $\bigcup \sigma_i = \mathrm{R}^n$ and thus $\mathcal{M} := \{\sigma_i \mid \mathrm{int}(\sigma_i) \neq \emptyset\}$.

To provide an example of this process, we return to the case where $T = N_C$. Observe that

$$N_C = \bigcup_{F_i} \bigcup_{x \in F_i} \{x\} \times N_F = \bigcup_{F_i} F_i \times N_{F_i}.$$

Thus, we see that N_C is the union of the polyhedral convex sets

$$C_i := \{x + y \mid (x,y) \in F_i \times N_{F_i}\} = F_i + N_{F_i}.$$

It follows that the process described above yields the normal manifold.

Robinson [22, Proposition 2.4] proved that the normal manifold is a subdivision of R^n. However, in general, the collection of cells $\mathcal{M}$ generated by the above process is not a subdivision. This can be demonstrated by the following example. Let

$$\begin{aligned} C_1 &:= \left\{(x,0) \in \mathbf{R}^2 \times \mathbf{R}^2 \mid x_1 \leq 0\right\} \\ C_2 &:= \left\{(x,0) \in \mathbf{R}^2 \times \mathbf{R}^2 \mid x_1 \geq 0, x_2 \geq 0\right\} \\ C_3 &:= \left\{(x,0) \in \mathbf{R}^2 \times \mathbf{R}^2 \mid x_1 \geq 0, x_2 \leq 0\right\} \end{aligned}$$

and let $T := \bigcup_{i=1}^3 C_i$. Observe that T is simply the zero mapping, and is thus a maximal monotone multifunction. However, employing our procedure for constructing $\mathcal{M}$, we obtain $\sigma_1 = \left\{x \in \mathrm{R}^2 \mid x_1 \leq 0\right\}$, $\sigma_2 = \left\{x \in \mathrm{R}^2 \mid x_1 \geq 0, x_2 \geq 0\right\}$, $\sigma_3 = \left\{x \in \mathrm{R}^2 \mid x_1 \geq 0, x_2 \leq 0\right\}$. Since $\sigma_1 \bigcap \sigma_2$ is not a face of σ_1, we see that $\mathcal{M} := \{\sigma_1, \sigma_2, \sigma_3\}$ is not a subdivision of $\mathbf{R}^n$.

Since P_T is single-valued, then by Lemma 3.3, P_T is affine on each cell $\sigma_i \in \mathcal{M}$. A representation of A_T on each cell is then given by (9). In order to have a workable description of these affine maps, it would appear necessary to exploit the underlying structure of T. One such case is the subject of the next section.

5 Separable T

A particularly important class of affine variational inequalities is that for which the set C is *rectangular*, i.e., C is defined by the constraints

$$l \leq z \leq u$$

where l and u are vectors in $\mathbf{R}^n$, with $l_i \in [-\infty, \infty)$ and $u_i \in (-\infty, \infty]$ for $1 \leq i \leq n$. This problem class has a number of features that are very attractive for pivotal algorithms similar to Algorithm AGE. In particular, the cells of linearity of the normal map are rectangular, and furthermore the normal map itself takes on a very simple form. Specifically, for an affine function $F(z) := Az + b$, the matrix used to represent the normal map on any cell is formed simply by replacing some of the columns of A by the corresponding columns of the identity matrix.

Rectangular variational inequalities are also attractive from a theoretical standpoint. In particular, if at least one of l_i and u_i is finite for each i, then the normal map is coherently oriented with respect to C if and only if A is a *P-matrix.*

Definition 5.1 ([4]) *A matrix A is said to be a P-*matrix *if all its principal minors are positive.*

Note that when C is rectangular, then $N_C(z) = N_{[l,u]}(z) = \prod_{i=1}^n N_{[l_i,u_i]}(z_i)$. This suggests that we can extend the notion of rectangularity to generalized equations by requiring that the multifunction T be *separable*, i.e., it is of the form

$$T(z) = \begin{bmatrix} T_1(z_1) \\ T_2(z_2) \\ \vdots \\ T_n(z_n) \end{bmatrix}.$$

where for each i, T_i is a maximal monotone polyhedral multifunction from R to R. With such a T, we shall see that the cells of linearity of the T-map A_T are rectangular.

We begin by looking at the resolvent operator $P_T = (I + T)^{-1}$. Note that

$$P_T(x) = \begin{bmatrix} P_{T_1}(x_1) \\ P_{T_2}(x_2) \\ \vdots \\ P_{T_n}(x_n) \end{bmatrix},$$

where for each i, $P_{T_i} = (1+T_i)^{-1}$. Since P_T is a continuous piecewise affine function, it follows that P_{T_i} is a a continuous piecewise affine function from R into R. Let k_i be the number of breakpoints of P_{T_i}. Then, for some strictly increasing sequence of breakpoints $\{\xi_{ij}\}$, $j = 1, \ldots, k_i$ and some set of coefficients $\{d_{ij}, b_{ij}\}$, $j = 0, \ldots, k_i$,

$$P_{T_i}(x) = \begin{cases} d_{i0}x + b_{i0} & x \leq \xi_{i1} \\ d_{ij}x + b_{ij} & \xi_{ij} \leq x \leq \xi_{i(j+1)}, \quad 1 \leq j < k_i \\ d_{ik_i}x + b_{ik_i} & \xi_{ik_i} \leq x. \end{cases}$$

Note that since P_T is monotonic and nonexpansive, $0 \leq d_{ij} \leq 1$.

The breakpoint sequence defines a subdivision of R given by $\mathcal{M}_i = \bigcup_{j=0}^{k_i} \sigma_{ij}$, where

$$\sigma_{ij} = \left\{ x \;\middle|\; \begin{array}{cc} x \leq \xi_{i1} & j = 0 \\ \xi_{ij} \leq x \leq \xi_{i(j+1)} & 0 < j < k_i \\ \xi_{ik_i} \leq x & j = k_i \end{array} \right\}.$$

We then define a subdivision of R^n by $\mathcal{M} = \prod_{i=1}^n \mathcal{M}_i$ with n-cells defined by $\sigma_{[j_1,j_2,\ldots,j_n]} = \sigma_{1j_1} \times \sigma_{2j_2} \times \ldots \times \sigma_{nj_n}$.

Clearly, P_T (and therefore A_T) is $\mathcal{M}$-PA. This establishes our earlier statement that the cells of linearity of A_T are rectangular. For each cell $\sigma_{[j_1,j_2,\ldots,j_n]}$ of $\mathcal{M}$, define a diagonal matrix $D_{[j_1,j_2,\ldots,j_n]}$ by $D_{[j_1,j_2,\ldots,j_n]}(i,i) = d_{ij_i}$. Further, define the vector $b_{[j_1,j_2,\ldots,j_n]} = [b_{1j_1}; b_{2j_2}; \ldots; b_{nj_n}]$. Then on $\sigma_{[j_1,j_2,\ldots,j_n]}$, P_T is represented by $(D_{[j_1,j_2,\ldots,j_n]}, b_{[j_1,j_2,\ldots,j_n]})$. Thus, on $\sigma_{[j_1,j_2,\ldots,j_n]}$, the T-map is given by

$$\begin{aligned} A_T(x) &= A(P_T(x)) + a + x - P_T(x) \\ &= (AD_{[j_1,j_2,\ldots,j_n]} + I - D_{[j_1,j_2,\ldots,j_n]})x + (A - I)b_{[j_1,j_2,\ldots,j_n]} + a \\ &= [d_{1j_1}A_{\cdot 1} + (1 - d_{1j_1})I_{\cdot 1}, d_{2j_2}A_{\cdot 2} + (1 - d_{2j_2})I_{\cdot 2}, \ldots, d_{nj_n}A_{\cdot n} \\ &\qquad + (1 - d_{nj_n})I_{\cdot n}]\, x + (A - I)b_{[j_1,j_2,\ldots,j_n]} + a \\ &=: M_{[j_1,j_2,\ldots,j_n]}x + \hat{b}_{[j_1,j_2,\ldots,j_n]} \end{aligned}$$

Thus, we see that the matrix $M_{[j_1,j_2,\ldots,j_n]}$ which represents A_T on $\sigma_{[j_1,j_2,\ldots,j_n]}$ has columns which are convex combinations of columns of A and the corresponding columns of I.

We now set about proving the main result of this section. Namely, if A is a P-matrix, then A_T is coherently oriented for any separable polyhedral maximal monotone multifunction T. We first need to prove two technical lemmas.

Lemma 5.2 *If A and B are $n \times n$ matrices where B is rank-1 such that* $\det(A) > 0$ *and* $\det(A + B) > 0$*, then* $\det(A + \lambda B) > 0$ *for all* $\lambda \in [0, 1]$.

Proof

$$\det(A + \lambda B) = \sum_{(j_1,\ldots,j_n)} \det[C_{j_1}, \ldots, C_{j_n}],$$

where the summation is taken over all possible choices of $(j_1, \ldots, j_n)$ such that C_{j_i} is either $A_{\cdot i}$ or $\lambda B_{\cdot i}$. Since B is rank-1, the determinants in the above sum are zero for all choices that include at least two columns of λB. Thus,

$$\begin{aligned} \det(A + \lambda B) &= \det A + \textstyle\sum_{i=1}^n \det[A_{\cdot 1}, \ldots, \lambda B_{\cdot i}, \ldots, A_{\cdot n}] \\ &= \det A + \lambda \left(\textstyle\sum_{i=1}^n \det[A_{\cdot 1}, \ldots, B_{\cdot i}, \ldots, A_{\cdot n}]\right) \end{aligned}$$

Thus, $\det(A + \lambda B)$ is an affine function of λ, which is positive at $\lambda = 0$ and $\lambda = 1$. Thus, it is positive for all $\lambda \in [0, 1]$. $\square$

Lemma 5.3 *Let A be an $n \times n$ matrix and let $\{B^1, \ldots, B^k\}$ be a collection of rank-1 $n \times n$ matrices. If* $\det(A + \lambda_1 B^1 + \cdots + \lambda_k B^k) > 0$ *for all choices of $\lambda_i = 0$ or 1, then* $\det(A + \lambda_1 B^1 + \cdots + \lambda_k B^k) > 0$ *for all choices of $\lambda_i \in [0, 1]$.*

Proof (By induction). The lemma is true for $k = 1$ by Lemma 5.2. Now, suppose the lemma is true for all $k < m$, we shall prove the lemma true for $k = m$.

Suppose $\{B^1, \ldots, B^m\}$ is a collection of rank-1 $n \times n$ matrices such that $\det(A + \lambda_1 B^1 + \cdots + \lambda_m B^m) > 0$ for all choices of $\lambda_i = 0$ or 1. Let $\hat{A} := A + B^m$. Then $\hat{A}$ and $\{B^1, \ldots, B^{m-1}\}$ satisfy the conditions of the lemma for $k = m - 1$. Thus, if $\lambda_i \in [0, 1]$, $\forall i$, then

$$\det((A + B^m) + \lambda_1 B^1 + \cdots + \lambda_{m-1} B^{m-1}) > 0.$$

Similarly, with $\hat{A} := A$, we have

$$\det(A + \lambda_1 B^1 + \cdots + \lambda_{m-1} B^{m-1}) > 0.$$

From these two results, we see that if we let $\tilde{A} := A + \lambda_1 B^1 + \cdots + \lambda_{m-1} B^{m-1}$, then $\tilde{A}$ and B^m satisfy the hypotheses for Lemma 5.2. Thus, for $\lambda_m \in [0, 1]$

$$\begin{aligned} 0 &< \det(\tilde{A} + \lambda_m B^m), \\ &= \det(A + \lambda_1 B^1 + \cdots + \lambda_m B^m). \end{aligned}$$

□

Theorem 5.4 *If A is a P-matrix, then for* any *separable maximal monotone polyhedral multifunction T, the T-map A_T defined by (9) has the property that in any cell of linearity, the matrix representing A_T has positive determinant. In particular, A_T is coherently oriented.*

Proof Let A_T have the representation $(\hat{A}, b)$ in the n-cell σ. By the our earlier discussion, $\hat{A}$ can be formed by replacing columns of A by a convex combination of columns of A and the corresponding columns of the identity matrix. Thus, the matrix is of the form

$$\hat{A} = A + \lambda_1 B^1 + \cdots + \lambda_n B^n, \qquad \lambda_i \in [0, 1]$$

where $B^i := (I_{\cdot i} - A_{\cdot i}) I_{\cdot i}^{\top}$. Observe that B^i is a rank-1 matrix.

Since A is a P-matrix, the matrix formed by replacing an arbitrary set of columns of A by corresponding columns of the identity matrix has positive determinant. Thus, the matrices $A, B^1, \ldots, B^n$ satisfy the hypotheses of Lemma 5.3. Thus, by Lemma 5.3, $\det(\hat{A}) > 0$. □

Corollary 5.5 *If A is a P-matrix and T is a separable maximal monotone polyhedral multifunction, then using lexicographic ordering, Algorithm AGE will find a solution to $A_T(x) = 0$ in a finite number of steps.*

6 Piecewise Linear-Quadratic Programming

We conclude by giving an example of a well known problem in mathematical programming that can be solved using the technique we have presented. The piecewise linear-quadratic programming problem (PLQP) is given by

$$\min\ h(x) = f(x) + \phi(Ax), \tag{10}$$

where $A \in \mathbf{R}^{m \times n}$, and $f : \mathbf{R}^n \to \mathbf{R} \bigcup \{\infty\}$ and $\phi : \mathbf{R}^m \to \mathbf{R} \bigcup \{\infty\}$ are convex *piecewise linear-quadratic* functions, defined below.

Definition 6.1 *A function $f : \mathbf{R}^n \to \mathbf{R} \bigcup \{\infty\}$ is* piecewise linear-quadratic *if* $\operatorname{dom} f$ *is closed and convex and there exists a finite subdivision $\mathcal{M}$ of* $\operatorname{dom}(f)$ *such that for each $\sigma \in \mathcal{M}$, $f_{|\sigma}$ is a quadratic function.*

Note that dom$f$ is polyhedral, and further that since the cells in the subdivision are closed, f is a continuous function on dom$f$.

The optimality conditions for PLQP are stated by the relation

$$0 \in \partial h(x),$$

where ∂h is the *convex subdifferential* operator defined by

$$\partial h(x) := \{z | h(w) \geq h(x) + z^\top (w - x), \forall w \in \mathrm{dom}(h)\}.$$

Under an appropriate constraint qualification (i.e. $\mathrm{ri}(A(\mathrm{dom}(f))) \bigcap \mathrm{ri}(\mathrm{dom}(\phi)) \neq \emptyset$), it follows that

$$\partial h(x) = \partial f(x) + A^\top \partial \phi(Ax).$$

Thus, for the optimality conditions to be satisfied, there must be an $x \in \mathrm{dom}(f)$ and $y \in \partial \phi(Ax)$ such that $-A^\top y \in \partial f(x)$. By [23, Theorem 23.5], the first statement is equivalent to

$$Ax \in \partial \phi^*(y),$$

where ϕ^* is the conjugate of ϕ. The optimality conditions are then

$$\begin{array}{rcl} -A^\top y & \in & \partial f(x) \\ Ax & \in & \partial \phi^*(y). \end{array}$$

Thus, if we define

$$\hat{A} := \begin{bmatrix} 0 & A^\top \\ -A & 0 \end{bmatrix}, \qquad T(x;y) := \begin{bmatrix} \partial f(x) \\ \partial \phi^*(y) \end{bmatrix},$$

the optimality conditions for PLQP can stated as the generalized equation

$$-\hat{A}(x;y) \in T(x;y). \tag{11}$$

The fact that T is polyhedral was shown in [25]. Thus, the optimality conditions for the piecewise linear-quadratic program can be expressed as an affine generalized equation, which can then be solved using our algorithm.

7 Acknowledgement

The authors would like to thank Stephen Robinson for constructive comments, suggestions and pertinent references that greatly improved the presentation of the material given in this paper.

References

[1] H. Brézis. *Opérateurs Maximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hilbert*. North-Holland, 1973.

[2] M. Cao and M. C. Ferris. P_C matrices and the linear complementarity problem. *Linear Algebra and Its Applications, forthcoming*, 1996. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

[3] M. Cao and M. C. Ferris. A pivotal method for affine variational inequalities. *Mathematics of Operations Research*, 21:44–64, 1996.

[4] R. W. Cottle, J. S. Pang, and R. E. Stone. *The Linear Complementarity Problem.* Academic Press, Boston, 1992.

[5] S. P. Dirkse. *Robust Solution of Mixed Complementarity Problems.* PhD thesis, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1994. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

[6] 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.

[7] S. P. Dirkse and M. C. Ferris. A pathsearch damped Newton method for computing general equilibria. *Annals of Operations Research, forthcoming*, 1996.

[8] B. C. Eaves. On the basic theorem of complementarity. *Mathematical Programming*, 1:68–87, 1971.

[9] B. C. Eaves. A short course in solving equations with PL homotopies. In R. W. Cottle and C. E. Lemke, editors, *Nonlinear Programming*, pages 73–143, Providence, RI, 1976. American Mathematical Society, SIAM–AMS Proceedings.

[10] B. C. Eaves. Computing stationary points. *Mathematical Programming Study*, 7:1–14, 1978.

[11] B. C. Eaves. Computing stationary points, again. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 3*, pages 391–405. Academic Press, New York, 1978.

[12] B. C. Eaves and C. E. Lemke. Equivalence of LCP and PLS. *Mathematics of Operations Research*, 6:475–484, 1981.

[13] M. C. Ferris and J. S. Pang. Engineering and economic applications of complementarity problems. Discussion Papers in Economics 95–4, Department of Economics, University of Colorado, Boulder, Colorado, 1995. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

[14] J. F. P. Hudson. *Piecewise Linear Topology.* Benjamin, New York, 1969.

[15] S. Karamardian. Generalized complementarity problem. *Journal of Optimization Theory and Applications*, 8:161–167, 1971.

[16] C. E. Lemke and J. T. Howson. Equilibrium points of bimatrix games. *SIAM Journal on Applied Mathematics*, 12:413–423, 1964.

[17] G. J. Minty. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematics Journal*, 29:341–346, 1962.

[18] D. Ralph. Global convergence of damped Newton's method for nonsmooth equations, via the path search. *Mathematics of Operations Research*, 19:352–389, 1994.

[19] S. Robinson. Some continuity properties of polyhedral multifunctions. *Mathematical Programming Study*, 14:206–214, 1979.

[20] S. M. Robinson. Generalized equations and their solution: Part I: Basic theory. *Mathematical Programming Study*, 10:128–141, 1979.

[21] S. M. Robinson. Generalized equations. In A. Bachem, M. Grötchel, and B. Korte, editors, *Mathematical Programming: The State of the Art, Bonn 1982*, pages 346–367. Springer Verlag, Berlin, 1983.

[22] S. M. Robinson. Normal maps induced by linear transformations. *Mathematics of Operations Research*, 17:691–714, 1992.

[23] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, New Jersey, 1970.

[24] T. F. Rutherford. MILES: A mixed inequality and nonlinear equation solver. Working Paper, Department of Economics, University of Colorado, Boulder, 1993.

[25] J. Sun. *On Monotropic Piecewise Quadratic Programming.* PhD thesis, University of Washington, Seattle, Washington, 1986.

[26] M. J. Todd. A note on computing equilibria in economies with activity analysis models of production. *Journal of Mathematical Economics*, 6:135–144, 1976.