JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 40, No. 3, JULY 1983

Method of Reduction in Convex Programming[1]

HENRY WOLKOWICZ[2]

Communicated by A. V. Fiacco

Abstract. We present an algorithm which solves a convex program with faithfully convex (not necessarily differentiable) constraints. While finding a feasible starting point, the algorithm reduces the program to an equivalent program for which Slater's condition is satisfied. Included are algorithms for calculating various objects which have recently appeared in the literature. Stability of the algorithm is discussed.

Key Words. Convexity, subdifferentials, cones of directions of constancy, equality set of constraints, stability.

1. Introduction

Consider the convex programming problem

$$\text{(P)} \qquad \text{minimize } f^\circ(x),$$

$$\text{subject to } f^k(x) \leq 0, \qquad k \in \mathcal{P} = \{1, \ldots, m\},$$

where $f^k: R^n \to R$, $k \in \{0\} \cup \mathcal{P}$ are convex functions. There are many algorithms in the literature that solve (P); see Refs. 1 and 2. These algorithms usually require that some constraint qualification holds at the optimal solution. Recently, Ben-Israel, Ben-Tal, and Zlobec (Ref. 3) have presented a characterization of optimality which does not require any constraint qualification. Algorithms based on this characterization were then given in Ref. 4. Further simplified characterizations were presented in Refs. 5, 6, 7, and 8. In this paper, we present the method of reduction, which solves (P) when the constraints are faithfully convex. This algorithm is based on the simplified characterizations and does not require any constraint qualification. In addition, the algorithm essentially extends *any*

[1] This research was partially supported by Grant No. NSERC-A3388.

[2] Associate Professor, Department of Mathematics, University of Alberta, Edmonton, Alberta, Canada.

0022-3239/83/0700-0349$03.00/0

algorithm that works when Slater's condition is satisfied. By a reduction process, the algorithm restricts the functions to smaller and smaller linear manifolds containing the feasible set. The smallest such linear manifold and a feasible point are obtained simultaneously. Slater's condition is satisfied for the reduced program, which then involves fewer variables and fewer constraints. Similar reductions for general convex programs and their dual properties are studied in Abrams and Wu (Ref. 9).

In Section 2, we present several preliminary definitions and results. Section 3 discusses the calculation of the cone of directions of constancy of a faithfully convex function h and introduces the notion of numerical rank for h. Section 4 finds the *equality set* of constraints. This is the set of constraints which are implicitly equality constraints, though explicitly written as inequality constraints. The algorithm given here is specifically designed for faithfully convex functions and differs from the ones given in Refs. 4 and 6. We also introduce the notion of numerical rank for the program (P). Section 5 presents the method of reduction. This algorithm combines the algorithms in the previous two sections to find a feasible point for (P) and simultaneously regularize (P). We conclude with a discussion on the stability of the algorithm. The numerical rank of (P) enables one to find a *stable estimate* of the solution of (P). This estimate is stable with respect to small perturbations, though the original solution might not be.

Note that the usual approach to solving convex programs (P), for which Slater's condition is not satisfied, is to positively perturb the right-hand sides of the constraint so as to satisfy Slater's condition; i.e., replace

$$f^k(x) \leq 0$$

by

$$f^k(x) \leq \epsilon, \qquad \epsilon > 0.$$

However, the solution of this perturbed program, denoted $\mu(\epsilon)$, converges to the solution of the dual to (P) as $\epsilon \downarrow 0$. Moreover, for ϵ small, the feasible set is very *thin*, which creates problems in numerical algorithms such as feasible direction methods. In particular, if we attempt to use the usual technique to avoid jamming (zigzagging), we must negatively perturb the right-hand sides of the constraints, thus negating the positive perturbation. The algorithm proposed here does the reverse. If Slater's condition is satisfied, but the feasible set is thin, as possibly obtained by perturbing (P), then the algorithm reduces the program to an affine manifold *parallel* to the feasible set and of a smaller dimension. The new feasible set is no longer thin, since it is considered in a space of smaller dimension, and it is an approximation to the original feasible set. See the discussions at the end of Section 4 and in Section 6.

2. Preliminaries

We consider the convex programming problem

$$\text{(P)} \qquad f^{\circ}(x) \to \min,$$

$$\text{s.t. } f^k(x) \le 0, \qquad k \in \mathcal{P} = \{1, \ldots, m\},$$

where $f^k: R^n \to R$ are convex functions, for all $k \in \{0\} \cup \mathcal{P}$. Without loss of generality, we assume that none of the functions is constant. The *feasible set* of (P) is

$$S = \{x \in R^n : f^k(x) \le 0, \text{for all } k \in \mathcal{P}\}.$$

The set of *binding constraints* at $x \in S$ is

$$\mathcal{P}(x) = \{k \in \mathcal{P} : f^k(x) = 0\}.$$

An important subset of $\mathcal{P}$, independent of x, is the *equality set*

$$\mathcal{P}^{=} = \{k \in \mathcal{P} : f^k(x) = 0, \text{ for all } x \in S\}.$$

See, e.g., Abrams and Kerzner (Ref. 5). This is the set of indices k for which the constraint f^k vanishes on the entire feasible set. We then denote

$$\mathcal{P}^{<}(x) = \mathcal{P}(x) \backslash \mathcal{P}^{=}.$$

Note that, unlike $\mathcal{P}^{=}$, $\mathcal{P}^{<}(x)$ depends on x. *Slater's condition* holds for (P) if there exists $\hat{x} \in S$, such that

$$f^k(\hat{x}) < 0, \qquad \text{for all } k \in \mathcal{P}.$$

This is equivalent to $\mathcal{P}^{=} = \varnothing$.

Following Ben-Tal, Ben-Israel, and Zlobec (Ref. 3), we define the *cone of directions of constancy* of f at x by

$$D_f^{=}(x) = \{d \in R^n : \text{there exists } \bar{\alpha} > 0, \text{with } f(x + \alpha d) = f(x), \text{for all } 0 < \alpha \le \bar{\alpha}\}.$$

For simplicity of notation, we let

$$D_k^{=}(x) = D_{f^k}^{=}(x),$$

$$D_{\Omega}^{=}(x) = \bigcap_{k \in \Omega} D_k^{=}(x), \qquad \text{for } \Omega \subset \mathcal{P}.$$

Remark 2.1. Following Rockafellar (Ref. 10), we say that a convex function f is *faithfully convex* if the following condition is satisfied: f is affine on a line segment only if it is affine on the whole line containing that segment. For a function f in the class of faithfully convex functions, the cone $D_f^{=}(x)$ is a subspace independent of x. Moreover, Rockafellar has shown that f is faithfully convex if and only if it is of the form

$$f(x)=h(Ax+b)+a\cdot x+\alpha, \tag{1}$$

where

$$A\in R^{m\times n},\qquad b\in R^m,\qquad a\in R^n, \alpha\in R,$$

and the function $h:R^m\to R$ is strictly convex. It is easy to see that

$$D_f^{=}(x)=\mathcal{N}\begin{pmatrix}A\\ a^t\end{pmatrix}$$

and is a subspace independent of x. The symbol $\mathcal{N}(\cdot)$ denotes null space and the superscript t denotes transpose.

We now recall some concepts dealing with directional derivatives and subgradients of a convex function f.

The *directional derivative* of f at x in the direction d is defined as

$$\nabla f(x;d)=\lim_{t\downarrow 0}[f(x+td)-f(x)]/t.$$

Convex functions have the useful property that the directional derivatives exist universally (e.g., Ref. 11, Theorem 23.1).

A vector $\phi\in R^n$ is said to be a *subgradient* of a convex function f at the point x, if

$$f(z)\geq f(x)+\phi\cdot(z-x),\qquad \text{for all } z\in R^n,$$

where $\phi\cdot(z-x)$ denotes the dot product in R^n. The set of all subgradients of f at x is then called the *subdifferential* of f at x and is denoted by $\partial f(x)$. If f is differentiable at x, then the *gradient* of f at x is denoted $\nabla f(x)$. Note that, in this case,

$$\partial f(x)=\{\nabla f(x)\}.$$

A useful relationship between the subdifferential and the directional derivative is (e.g., Ref. 11)

$$\nabla f(x;d)=\max\{\phi\cdot d:\phi\in\partial f(x)\}. \tag{2}$$

For every subset Ω of $\mathcal{P}(x)$, the linearizing cone at $x\in S$, with respect to Ω, is

$$C_\Omega(x)=\{d\in R^n:\phi\cdot d\leq 0, \text{for all } \phi\in\partial f^k(x) \text{ and all } k\in\Omega\}.$$

By (2), we see that

$$C_\Omega(x)=\{d\in R^n:\nabla f^k(x;d)\leq 0, \text{for all } k\in\Omega\}.$$

The *cone of subgradients* at x is

$$B_\Omega(x) = \{\phi \in R^n : \phi = \sum_{k\in\Omega} \lambda_k \phi^k, \text{ for some } \lambda_k \geq 0 \text{ and } \phi^k \in \partial f^k(x)\}.$$

We set

$$B_\varnothing(x) = \{0\}.$$

Recall that, for $M \subset R^n$ the *polar* of M is

$$M^* = \{\phi \in R^n : \phi \cdot x \geq 0, \text{ for all } x \in M\}.$$

M^* is then a closed convex cone in R^n. Furthermore, if $K, L \subset R^n$, then

$$K^{**} = \overline{\text{cone}\, K}, \tag{3}$$

the closure of the convex cone generated by K, while

$$(K \cap L)^* = \overline{K^* + L^*}. \tag{4}$$

The linearizing cone and the cone of subgradients have the following dual property.

Lemma 2.1. Suppose that $\Omega \subset \mathcal{P}$. Then,

$$\overline{B_\Omega(x)} = -C^*_\Omega(x).$$

Proof. Since, by (3) and by definition,

$$B_k(x) = \partial f^k(x)^{**} = -C^*_k(x), \tag{5}$$

we conclude [by (4) and (5)] that

$$-C^*_\Omega(x) = \overline{-\sum_{k\in\Omega} C^*_k(x)} = \overline{B_\Omega(x)}. \qquad \square$$

Gould and Tolle (Ref. 12) used Farkas' lemma to prove the above results for differentiable functions on R^n. Note that, in this case, $B_\Omega(x)$ is finitely generated and therefore closed. In addition, $B_\Omega(x)$ is closed when

$$0 \notin \text{conv} \bigcup_{k\in\Omega} \partial f^k(x),$$

since it is then a compactly generated cone; see, e.g., Holmes, Ref. 13.

We will also need the following two theorems of the alternative.

Theorem 2.1. (*Dubovitskii and Milyutin, Ref. 14*). Let $C_1, \ldots, C_k$ be open (blunt) convex cones, and let C_{k+1} be a convex cone. Then,

$$\bigcap_{i=1}^{k+1} C_i = \varnothing,$$

if and only if there exist vectors

$$y_i \in C_i^*, \qquad i=1,\ldots,k+1,$$

not all zero, such that

$$\sum_{i=1}^{k+1} y_i = 0.$$

Theorem 2.2. *Motzkin's Theorem of the Alternative (Refs. 1 and 15).* Let

$$A_i \in R^{k\times l_i}, \qquad i=1,2,3, \qquad A_1 \neq 0.$$

Then, exactly one of the following two systems is consistent:

(I) $A_1x^1 + A_2x^2 + A_3x^3 = 0, 0 \neq x^1 \geq 0, x^2 \geq 0;$

(II) $A_1^t y > 0, A_2^t y \geq 0, A_3^t y = 0.$

The algorithm is based on the following regularization technique.

Theorem 2.3. Let $\bar{x} \in S$ and f^k, $k \in \mathscr{P}^=$, be faithfully convex. Suppose that the $n \times r$ matrix A satisfies

$$D_{\mathscr{P}^=}^{=} = \mathscr{R}(A),$$

the range space of A. Now, consider the program, in the variable $y \in R^r$,

$$(\mathrm{P_r}) \qquad f^\circ(\bar{x}+Ay) \to \min,$$
$$\text{s.t. } f^k(\bar{x}+Ay) \leq 0, \qquad k \in \mathscr{P}\backslash\mathscr{P}^=.$$

Then, Slater's condition is satisfied for $(\mathrm{P_r})$, and $y=0$ is a feasible point. Moreover, if y^* solves $(\mathrm{P_r})$, then $\bar{x}+Ay^*$ solves (P).

This theorem was presented in Refs. 16 and 17. The proof follows readily from the characterization of optimality given in Refs. 5 and 6. If the matrix A is chosen to be of full column rank and $\mathscr{P}^= \neq \varnothing$, then the program $(\mathrm{P_r})$ has less variables and less constraints than the original program (P). In fact,

$$r = \dim S.$$

3. Calculating the Cone of Directions of Constancy

Recall that the cone of directions of constancy $D_f^=$, of a faithfully convex function $f: R^n \to R$, is a subspace of R^n independent of the choice of x; see Remark 2.1. An algorithm for calculating $D_f^=$, when f is differentiable, is given in Ref. 18. The algorithm can be modified to use subdifferentials, rather than gradients (see Ref. 17).

The algorithm is based on the fact that $D_f^=$ lies in the orthogonal complement of ϕ, for any $\phi \in \partial f(x)$. By repeatedly considering the restriction of f to this orthogonal complement, we calculate $D_f^=$. We stop when

$$\partial f(Pe_i) = \{0\},$$

for the set of $k+1$ affinely independent points $e^i \in \{0\} \cup E_k$ under consideration. Then,

$$D_f^= = \mathcal{R}(P) \text{ or } \{0\},$$

depending on the current value of k. See flowchart in Fig. 1. (Note that $\phi \in \partial f(y)$ is taken as a row vector in the flowcharts.)

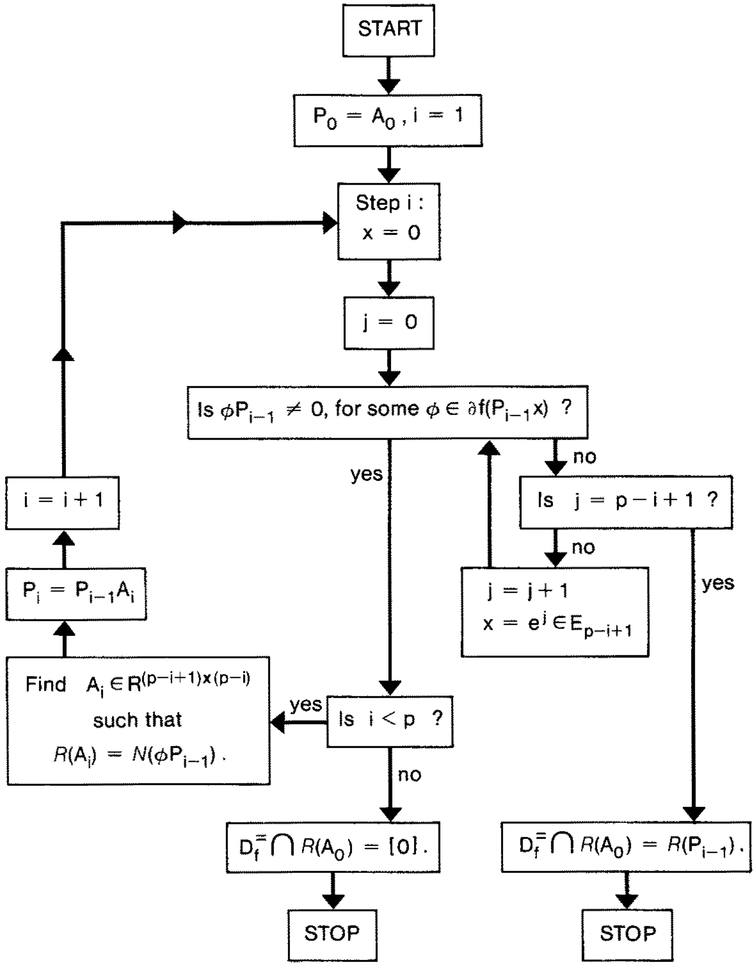

Fig. 1. Flowchart to find $D_f^= \cap \mathcal{R}(A_0)$.

Improvements in efficiency and accuracy have been made for this algorithm. However, it appears that we cannot overcome completely the question of stability. This is because a key step involves testing whether a variable is exactly zero, which is impossible on the computer. For example, to find $D_1^{\bar{=}}$ and $D_2^{\bar{=}}$ for the two functions

$$f^1(x) = 0.01x^2 \qquad \text{and} \qquad f^2(x) = 0.001x^2,$$

on a machine with two decimals accuracy, requires checking whether or not

$$\nabla f^i(0) = \nabla f^i(1) = 0, \qquad i = 1, 2.$$

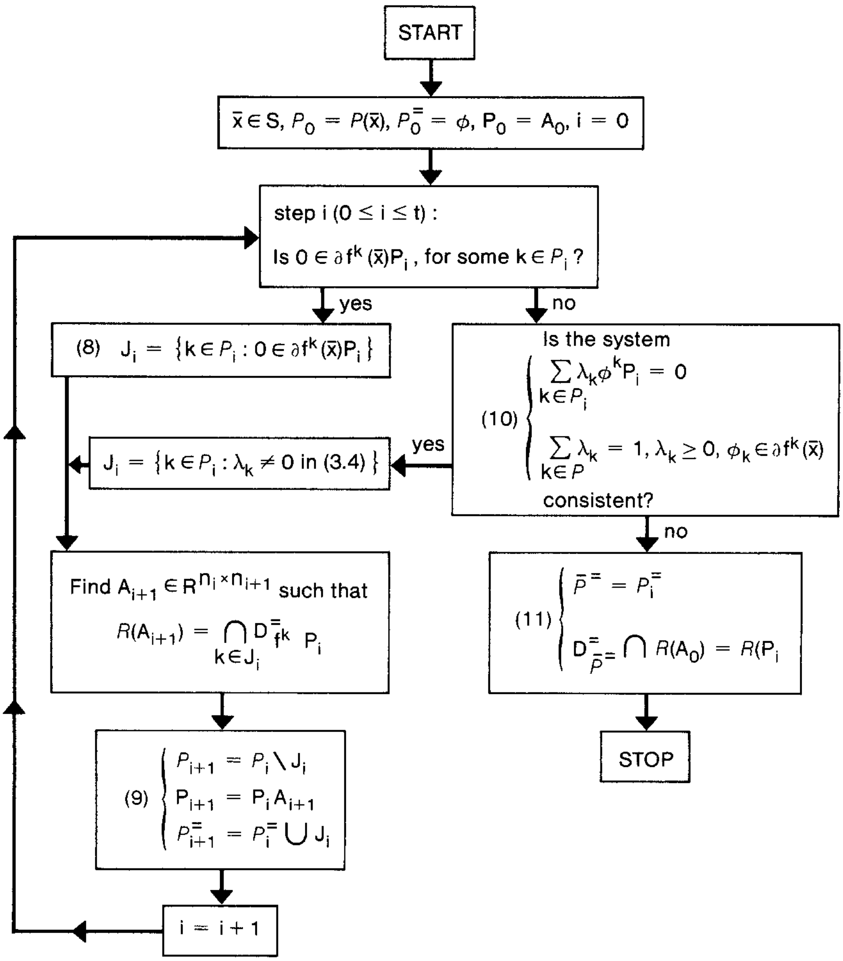

Fig. 2. Flowchart to find $\bar{\mathscr{P}}^=$ and $D_{\bar{\mathscr{P}}^=}^= \cap \mathscr{R}(A_0)$.

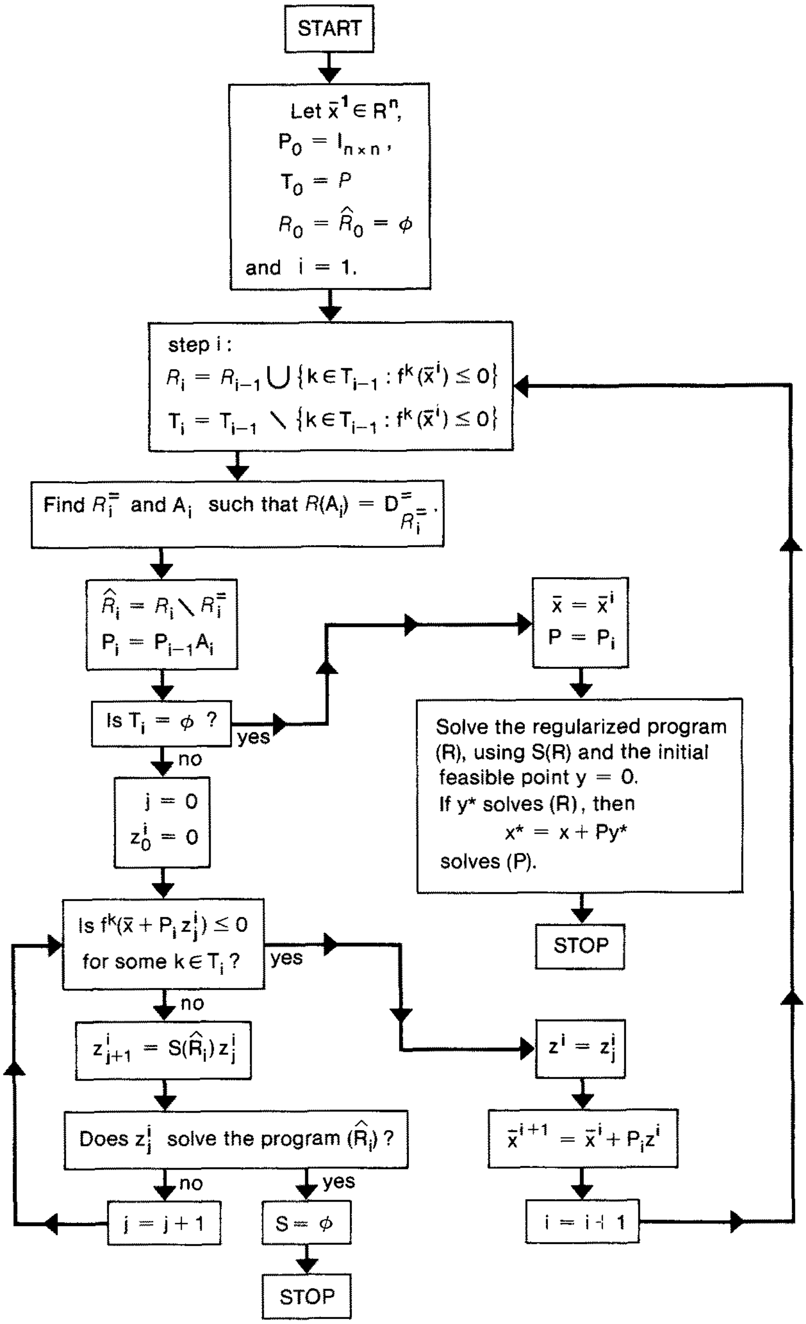

Fig. 3. Flowchart for the method of reduction.

We would then conclude wrongly that

$$D_1^{=}=\{0\},$$

while

$$D_2^{=}=R.$$

Essentially, we are finding an approximation for the second derivative. The instability is a result of the instability in numerical differentiation. We could improve our accuracy by checking the gradient at some point large than 1, or by improving the estimate of the second derivative analytically (if the second derivative is given analytically), but this does not completely remove the problem. Thus, we might calculate the subspace $D_f^{=}$ with a dimension larger than it actually has. However, if we consider a bounded region Ω, then we can conclude that

$$|f(y)-f(x)|<\epsilon, \qquad \text{for all } x, y \in \Omega \cap D_f^{=}, \tag{6}$$

where $D_f^{=}$ is calculated using the desired accuracy $\epsilon>0$. More precisely, if

$$\Omega \cap D_f^{=} \subset \operatorname{conv}\{Py_i\}_{i=1}^{k+1},$$

where conv denotes convex hull, and

$$\|Py_i\| \leq K, \qquad \text{for all } i,$$

then we require

$$\|\phi\| \leq \epsilon/2K, \qquad \text{for all } \phi \in \partial f(Py_i).$$

Moreover, given a desired tolerance $\epsilon>0$, we can similarly guarantee that, if $d \in D_f^{=}$, then

$$|f(d)-f(0)|<\epsilon, \qquad \text{if } \|d\| \leq K.$$

Thus, the algorithm numerically finds the cone of directions of *almost constancy*.

The question of calculating $D_f^{=}$ might be compared to the problem of finding $\mathcal{N}(A)$, the null space of a matrix A. This problem is unstable. We can define the *rank* of a faithfully convex function f as $n-\dim D_f^{=}$. Then, finding the rank of a faithfully convex function is unstable [and, as we shall see in the sequel, vital to the stability of solving (P)]. We can now define the numerical rank of f. This extends the notion of numerical rank for matrices given in Ref. 19. We let

$$\|f\|_t=\max\{|f(x)|: \|x\| \leq t\}.$$

This defines a seminorm on the space of continuous functions on R^n. It is a norm on the subspace of functions which are identically zero, whenever they are zero on the entire t-ball. This includes all faithfully convex functions.

Definition 3.1. A faithfully convex function has *numerical rank* (δ, ϵ, r), with respect to the norm $\|\cdot\|_t$, if

$$r = \inf\{\text{rank } g : g \text{ is faithfully convex and } \|f-g\|_t < \epsilon\},$$

$$\epsilon < \delta \leq \sup\{\eta : g \text{ faithfully convex}, \|f-g\|_t < \eta \text{ implies rank } g \geq r\}.$$

The introduction of δ guarantees that the numerical rank does not decrease with small increases in ϵ, and thus is a stable number.

In summary, the algorithm theoretically finds $D_f^=$, the cone of directions of constancy. Numerically, we can guarantee only that it finds the cone of directions of *almost constancy*, to a given tolerance ϵ. The dimension of $D_f^=$ is numerically unstable. However, using the numerical rank, we can find a stable upper bound for the dimension of $D_f^=$, equivalently a stable lower bound for the numerical rank.

4. Calculating the Sets $\mathcal{P}^=$ and $D_{\mathcal{P}^=}^=$

In Ref. 5, an algorithm for calculating $\mathcal{P}^=$ is given for the program (P). We now present a modified version of this algorithm, in the case that the constraints f^k, $k \in \mathcal{P}^=$, are faithfully convex. In actual fact, the algorithm finds

$$\bar{\mathcal{P}}^= \text{ and } D_{\bar{\mathcal{P}}^=}^= \cap \mathcal{R}(A_0),$$

in at most t steps, with

$$t = \min\{\text{card } \mathcal{P}(\bar{x}), n+1-\dim[S \cap (\bar{x}+\mathcal{R}(A_0))]\},$$

$$\bar{\mathcal{P}}^= = \{k \in \mathcal{P} : f^k(x) = 0, \text{ for all } x \in S \cap (\bar{x}+\mathcal{R}(A_0))\},$$

and A_0 is any specified $n \times n_0$ matrix. Recall that $D_k^=$ is independent of x, when the function f^k is faithfully convex. If A_0 is specified to be the identity, then $\mathcal{P}^=$ and $D_{\mathcal{P}^=}^=$ are found. The generalization to find $\bar{\mathcal{P}}^=$ and $D_{\bar{\mathcal{P}}^=}^=$ will be needed in the sequel.

The algorithm is a (finite) iterative method. We start with

$$\mathcal{P}_0^= = \varnothing$$

and find the sets

$$\mathcal{P}_{i+1}^= = \mathcal{P}_i^= \cup J_i$$

at each iteration. The sets J_i are defined below. The algorithm terminates when

$$\mathcal{P}_i^{=} = \bar{\mathcal{P}}^{=}$$

is reached. The difference between this algorithm and the one in Ref. 5 is that, at each iteration, we discard the constraints f^k, $k \in J_i$; and, by a substitution technique, we then consider the remaining constraints as being restricted to the subspace $D^{=}_{J_i}$. In addition, when finding the set J_i, we first check if $\phi = 0$ is in the subdifferential of any of the (remaining) binding constraints. Recall that, if $0 \in \partial f(x)$ and f is convex, then f achieves a global minimum at x. The algorithm is demonstrated in Example 4.1 below. Note that $f^k 0 p_i$ denotes the composition of f^k and P_i, i.e.,

$$f^k 0 p_i(y) = f(P_i y).$$

Algorithm B

Initialization. Let

$$\bar{x} \in S, \qquad \mathcal{P}_0 = \mathcal{P}(\bar{x}), \qquad \mathcal{P}_0^{=} = \varnothing, \qquad P_0 = A_0, \qquad i = 0.$$

ith step, $0 \leq i \leq t$. Find $k \in \mathcal{P}_i$ such that

$$0 \in \partial f^k(\bar{x}) P_i.$$

Case (i). If such a k exists, use Algorithm A to find the $n_i \times n_{i+1}$ matrix A_{i+1} satisfying

$$\mathcal{R}(A_{i+1}) = \bigcap_{k \in J_i} D^{=}_{f^k_0 P_i}, \tag{7}$$

where

$$J_i = \{k \in \mathcal{P}_i \colon 0 \in \partial f^k(\bar{x}) P_i\}. \tag{8}$$

Then, set

$$\mathcal{P}_{i+1} = \mathcal{P}_i / J_i, \qquad P_{i+1} = P_i A_{i+1}, \qquad \mathcal{P}_{i+1}^{=} = \mathcal{P}_i^{=} \cup J_i, \tag{9}$$

and proceed to step $i+1$.

Case (ii). If such a k does not exist, but the system

$$\sum_{k \in \mathcal{P}_i} \lambda_k \phi^k P_i = 0,$$
$$\sum_{k \in \mathcal{P}_i} \lambda_k = \hat{1}, \qquad \lambda_k \geq 0, \qquad \phi^k \in \partial f^k(\bar{x}), \tag{10}$$

is consistent, then find A_{i+1}, P_{i+1}, $\mathcal{P}_{i+1}$, $\mathcal{P}_{i+1}^{=}$ satisfying (7) and (9), where

$$J_i = \{k \in \mathcal{P}_i \colon \lambda_k \neq 0 \text{ in } (10)\}.$$

Now, proceed to step $i+1$.

Case (iii). If such a k does not exist, but the system (10) is inconsistent, then stop.

We conclude that

$$\bar{\mathcal{P}} = \mathcal{P}_i^=, \qquad D^=_{\mathcal{P}^=} \cap \mathcal{R}(A_0) = \mathcal{R}(P_i). \tag{11}$$

Before proving the convergence of the algorithm, let us first prove the following rather technical lemma.

Lemma 4.1. Denote

$$f_i^k(y) \triangleq f^k(\bar{x} + P_i y), \qquad S_i \triangleq \{x \in R^{n_i}: f_i^k(x) \le 0, \text{ for all } k \in \mathcal{P}_i\}.$$

Then,

$$f_i^k(y) = f_{i-1}^k(A_i y), \tag{12}$$

$$D^=_{f_i^k} = D^=_{f^k \circ P_i}, \qquad \text{for all } k \in \mathcal{P}_i, \tag{13}$$

$$D^=_{\mathcal{P}_i^=} \cap \mathcal{R}(A_0) = \mathcal{R}(P_i), \tag{14}$$

$$S_i \subset \mathcal{R}(A_{i+1}), \tag{15}$$

$$S \cap (\bar{x} + \mathcal{R}(A_0)) \subset \bar{x} + \mathcal{R}(P_i). \tag{16}$$

Proof. Since

$$f_i^k(y) = f^k(\bar{x} + P_i y) = f^k(\bar{x} + P_{i-1} A_i y) = f_{i-1}^k(A_i y),$$

relation (12) is proved.

Now, when f^k is faithfully convex, there exists a strictly convex function g, a matrix A, vectors a and b, and a constant c such that

$$f^k(x) = g(Ax + b) + a^t x + c,$$

with

$$D^=_{f^k} = \mathcal{N}\begin{pmatrix} A \\ a^t \end{pmatrix};$$

see Remark 2.1. Therefore,

$$\begin{aligned} f_i^k(y) &= f^k(\bar{x} + P_i y) \\ &= g(A(\bar{x} + P_i y) + b) + a^t(\bar{x} + P_i y) + c \\ &= g(AP_i y + A\bar{x} + b) + a^t P_i y + a^t \bar{x} + c, \end{aligned}$$

with

$$D^=_{f_i^k} = \mathcal{N}\begin{pmatrix} AP_i \\ a^t P_i \end{pmatrix},$$

which equals the cone of directions of constancy of

$$f^k 0 P_i(y) = g(AP_i y + b) + a^t P_i y + c.$$

This proves (13).

Let us prove (14) by finite induction on i. The result holds for $i = 0$, since

$$\mathscr{P}_0^= = \varnothing \quad \text{and} \quad P_0 = A_0.$$

So, let us assume that $i \geq 1$ and that

$$D^{=}_{\mathscr{P}^=_{i-1}} \cap \mathscr{R}(A_0) = \mathscr{R}(P_{i-1}). \tag{17}$$

Note that we will consider $D_k^=$ as a subset of R^n and as a subset of R^{n_i} depending on the context, i.e., depending on whether we are considering the function f^k or f_i^k. First, suppose that $d \in \mathscr{R}(P_i)$, i.e.,

$$d = A_0 A_1 \cdots A_i \bar{y}, \qquad \text{for some } \bar{y} \in R^{n_i}.$$

This implies that

$$d \in \mathscr{R}(P_{i-1}) = D^{=}_{\mathscr{P}^=_{i-1}} \cap \mathscr{R}(A_0),$$

by (17). Now, to show that

$$d \in D^{=}_{\mathscr{P}^=_i} \cap \mathscr{R}(A_0),$$

it is sufficient to show that

$$d \in D^{=}_{\mathscr{P}^=_i \setminus \mathscr{P}^=_{i-1}} = D^{=}_{J_{i-1}},$$

by (9). So, let

$$k \in J_{i-1} \quad \text{and} \quad \alpha \in R.$$

Then, by (12),

$$f^k(\bar{x} + \alpha d) = f^k(\bar{x} + \alpha A_0 \cdots A_i \bar{y}) = f_{i-1}^k(\alpha A_i \bar{y});$$

since

$$k \in J_{i-1} \quad \text{and} \quad \mathscr{R}(A_i) \subset D^{=}_{f_{i-1}^k},$$

by (7) and (13), we have

$$f^k(\bar{x} + ad) = f_{i-1}^k(0) = f^k(\bar{x}).$$

This implies that

$$d \in D^{=}_{J_{i-1}}.$$

Thus, we have shown that

$$\mathscr{R}(P_i) \subset D^{=}_{\mathscr{P}^=_i}.$$

Conversely, suppose that

$$d \in D^{=}_{\mathscr{P}_i^=} \cap \mathscr{R}(A_0).$$

Since

$$\mathscr{P}_i^= \supset \mathscr{P}_{i-1}^=,$$

(17) implies that

$$d = A_0 A_1 \cdots A_{i-1}\bar{y}, \qquad \text{for some } \bar{y} \in R^{n_{i-1}}.$$

To show that

$$D^{=}_{\mathscr{P}_i^=} \cap \mathscr{R}(A_0) \subset \mathscr{R}(P_i),$$

it is now sufficient to show that

$$\bar{y} = A_i \bar{z}, \qquad \text{for some } \bar{z} \in R^{n_i}.$$

Suppose that

$$k \in \mathscr{P}_i^= \backslash \mathscr{P}_{i-1}^= = J_{i-1} \quad \text{and} \quad \alpha \in R.$$

Then, by (12) and since

$$d \in D^{=}_{\mathscr{P}_i^=} \quad \text{and} \quad k \in \mathscr{P}_i^=,$$

we have

$$f_{i-1}^k(0) = f^k(\bar{x}) = f^k(\bar{x} + \alpha d) = f^k(\bar{x} + A_0 \cdots A_{i-1}(\alpha \bar{y})) = f_{i-1}^k(\alpha \bar{y}).$$

This implies that

$$\bar{y} \in D^{=}_{J_{i-1}} = \mathscr{R}(A_i),$$

by (7). Thus,

$$\bar{y} = A_i \bar{z}, \qquad \text{for some } \bar{z} \in R^{n_i}.$$

This completes the proof of (14).

To prove (15), we consider two separate cases.

Case (a). Suppose that

$$0 \in \partial f_i^k(0), \qquad \text{for some } k \in \mathscr{P}_i.$$

Note that

$$\partial f_i^k(0) = \partial f^k(\bar{x}) P_i.$$

By (8),

$$0 \in \partial f_i^k(0), \qquad \text{for all } k \in J_i.$$

Therefore, $y=0$ is a global minimum for the convex functions f_i^k, $k \in J_i$. Now, suppose that $\bar{y} \in S_i$; i.e.,

$$f_i^k(\bar{y}) \le 0, \qquad \text{for all } k \in \mathscr{P}_i.$$

Then,

$$f_i^k(\bar{y}) = 0, \qquad \text{for all } k \in J_i,$$

since $y=0$ is a global minimum for these functions and

$$f_i^k(0) = f^k(\bar{x}) = 0, \qquad \text{for all } k \in \mathscr{P}_i \subset \mathscr{P}_0 \subset \mathscr{P}(\bar{x}).$$

Since S_i is convex and $0 \in S_i$, we conclude that

$$\alpha\bar{y} \in S_i, \qquad \text{for all } 0 \le \alpha \le 1.$$

This further implies that

$$f_i^k(\alpha\bar{y}) = 0, \qquad \text{for all } k \in J_i \text{ and } 0 \le \alpha \le 1,$$

i.e.,

$$\bar{y} \in D_{J_i}^{=} = \mathscr{R}(A_{i+1}).$$

This proves (15), in case (a).

Case (b). Suppose that

$$0 \notin \partial f_i^k(0), \qquad \text{for all } k \in \mathscr{P}_i.$$

Also, assume that the system (10) is consistent; i.e., there exist $\lambda_k > 0$ such that

$$\sum_{k \in J_i} \lambda_k \phi^k = 0, \qquad \phi^k \in \partial f_i^k(0). \tag{18}$$

Note that, if no such λ_k's exist, then the algorithm stops and (15) does not require proof. As in case (a), we need only show that

$$\text{if } \bar{y} \in S_i \text{ and } k \in J_i, \qquad \text{then } f_i^k(\bar{y}) = 0.$$

Suppose the contrary. Then, there exists

$$y \in S_i \quad \text{and} \quad k_0 \in J_i.$$

such that

$$f_i^k(y) \le 0, \qquad \text{for all } k \in \mathscr{P}_i, \qquad \text{and } f_i^{k_0}(y) < 0.$$

This implies that

$$\phi^k \cdot y \le 0, \qquad \text{and } \phi^{k_0} \cdot y < 0,$$

$$\text{for all } \phi^k \in \partial f_i^k(0),\ k \in J_i, \text{ and } \phi^{k_0} \in \partial f_i^{k_0}(0).$$

By Theorem 2.2, this contradicts (18). Therefore, (15) is proved.

Let us now prove (16), by finite induction on i. The result holds for $i=0$, since

$$P_0 = A_0.$$

So, let us assume that $i \geq 0$ and

$$S \cap (\bar{x} + \mathscr{R}(A_0)) \subset \bar{x} + \mathscr{R}(P_{i-1}).$$

Let

$$x \in S \cap (\bar{x} + \mathscr{R}(A_0)).$$

Then, the above implies that

$$x = \bar{x} + P_{i-1}\bar{y}, \qquad \text{for some } \bar{y} \in R^{n_{i-1}}.$$

Thus, by definition of f_{i-1}^k and since $x \in S$,

$$f_{i-1}^k(\bar{y}) = f^k(x) \leq 0.$$

Therefore,

$$\bar{y} \in S_{i-1}.$$

Now, by (15)

$$\bar{y} = A_i\bar{z}, \qquad \text{for some } \bar{z} \in R^{n_i}.$$

Substituting for $\bar{y}$ in the expression for x implies that

$$x = \bar{x} + A_0 \cdots A_i\bar{z},$$

which proves (16). Note that the sets $\bar{x} + \mathscr{R}(P_i)$ are decreasing linear manifolds containing the set $S \cap (\bar{x} + \mathscr{R}(A_0))$. The algorithm essentially stops when $\bar{x} + \mathscr{R}(P_i)$ is the smallest such linear manifold. □

We are now ready to prove the convergence of the algorithm. Recall that

$$\bar{P}^{=} = \{k \in \mathscr{P} : f^k(x) = 0, \text{ for all } x \in S \cap (\bar{x} + \mathscr{R}(A_0))\}.$$

Theorem 4.1. Suppose that $\bar{x} \in S$, A_0 is an arbitrary $n \times n_0$ matrix, and the constraints $f^k, k \in \bar{P}^{=}$ are faithfully convex. Then, the above algorithm finds

$$\bar{P}^{=} \quad \text{and} \quad D_{\bar{P}^{=}}^{=} \cap \mathscr{R}(A_0)$$

in at most t steps, where

$$t = \min\{\text{card}\, \mathscr{P}(\bar{x}), n_0 + 1 - \dim(S \cap (\bar{x} + \mathscr{R}(A_0)))\}.$$

Proof. We need to prove that (11) holds when case (iii) occurs. So, suppose that

$$0 \notin \partial f_i^k(0), \qquad \text{for all } k \in \mathscr{P}_i,$$

and the system (10) is inconsistent. Let

$$C_k \triangleq \text{int}\, C_{f_i^k}(0) = \{y \in R^{n_i}: \phi^k P_i y < 0, \text{ for all } \phi^k \in \partial f^k(\bar{x})\}.$$

Then, by Lemma 2.1 and since $0 \notin \partial f_i^k(0)$, we have that

$$C_k^* = \text{cone}\, \partial f^k(\bar{x}) P_i.$$

Since (10) is inconsistent, we conclude that the system

$$\sum_{k \in \mathscr{P}_i} y^k = 0, \qquad y^k \in \text{cone}\, \partial f^k(\bar{x}) P_i,$$

is inconsistent. Theorem 2.1 now implies that

$$\bigcap_{k \in \mathscr{P}_i} \{y \in R^{n_i}: \phi^k P_i \cdot y < 0, \text{ for all } \phi^k \in \partial f^k(\bar{x})\} \neq \varnothing.$$

This yields $\hat{y} \in R^{n_i}$, such that

$$\phi^k P_i \cdot \hat{y} < 0, \qquad \text{for all } k \in \mathscr{P}_i, \text{ and } \phi^k \in \partial f^k(\bar{x}). \tag{19}$$

Let

$$x(\alpha) = \bar{x} + P_i \alpha \hat{y}. \tag{20}$$

Then, (19) and (20) imply that

$$\begin{aligned} f^k(x(\alpha)) &< 0, \qquad \text{for all } k \in \mathscr{P} \backslash \mathscr{P}(\bar{x}), \\ f^k(x(\alpha)) &< 0, \qquad \text{for all } k \in \mathscr{P}_i, \end{aligned} \tag{21}$$

for all $0 \leq \alpha \leq \bar{\alpha}$, some $\bar{\alpha} > 0$. Furthermore, if $0 \leq \alpha \leq \bar{\alpha}$, then [By (20), (14), and since: $\mathscr{P}_i^= \subset \mathscr{P}(\bar{x})$],

$$\begin{aligned} f^k(x(\alpha)) &= f^k(\bar{x} + P_i \alpha \hat{y}), \\ &= f^k(\bar{x}), \qquad \text{for all } k \in \mathscr{P}_i^=, \\ &= 0, \qquad \text{for all } k \in \mathscr{P}_i^=. \end{aligned} \tag{22}$$

Therefore, (21) and (22) imply that

$$x(\alpha) \in S \cap (\bar{x} + \mathscr{R}(A_0));$$

moreover,

$$\bar{\mathscr{P}}^{=} \subset (\mathscr{P}(\bar{x}) \backslash \mathscr{P}_i) = \mathscr{P}_i^{=}.$$

Since

$$D_{\mathscr{P}_i^{=}}^{=} \cap \mathscr{R}(A_0) = \mathscr{R}(P_i),$$

by (14), to prove (11) we have only left to show that

$$\mathscr{P}_i^{=} \subset \bar{\mathscr{P}}^{=}. \tag{23}$$

Let us prove this by finite induction on i. Now, (23) holds for $i = 0$, since

$$\mathscr{P}_0^{=} = \varnothing.$$

Therefore, let us assume that $i \geq 1$ and

$$\mathscr{P}_{i-1}^{=} \subset \bar{\mathscr{P}}^{=}.$$

Since

$$\mathscr{P}_i^{=} = \mathscr{P}_{i-1}^{=} \cup J_{i-1},$$

by iteration, it is sufficient to show that

$$J_{i-1} \subset \bar{\mathscr{P}}^{=}.$$

Suppose the contrary. Then, there exists

$$x \in S \cap (\bar{x} + \mathscr{R}(A_0)) \quad \text{and} \quad k_0 \in J_{i-1},$$

such that

$$f^k(x) \leq 0, \qquad \text{for all } k \in \mathscr{P} \text{ and } f^{k_0}(x) < 0. \tag{24}$$

But

$$x = \bar{x} + A_0 \cdots A_i y, \qquad \text{for some } y \in R^{n_i},$$

by (16); and, by (12) and since

$$D_{J_{i-1}}^{=} = \mathscr{R}(A_i),$$

by (7) and (13), we have

$$f^{k_0}(x) = f^{k_0}(\bar{x} + A_0 \cdots A_i y) = f_{i-1}^{k_0}(A_i y) = 0.$$

This contradicts (24). □

Example 4.1. Suppose $S \subset R^5$ is defined by the constraints

$$f^1(x) = \exp(x_1) + x_2^2 - 1 \leq 0,$$
$$f^2(x) = x_1^2 + x_2^2 + \exp(-x_3) - 1 \leq 0,$$
$$f^3(x) = x_1 + x_4^2 + x_5^2 - 1 \leq 0,$$
$$f^4(x) = \exp(-x_2) - 1 \leq 0,$$
$$f^5(x) = (x_1 - 1)^2 + x_2^2 - 1 \leq 0,$$
$$f^6(x) = x_1 + \exp(-x_4) - 1 \leq 0,$$
$$f^7(x) = x_2 + \exp(-x_5) - 1 \leq 0.$$

Let us find $\mathcal{P}^=$ and $D^=_{\mathcal{P}^=}$.

Initialization. Let

$$\bar{x} = (0, 0, 1, (\tfrac{1}{2})\sqrt{2}, (\tfrac{1}{2})\sqrt{2})$$

be the chosen feasible point. Then,

$$P_0 = A_0 = I_{5\times 5}, \qquad \mathcal{P}_0 = \mathcal{P}(\bar{x}) = \{1, 3, 4, 5\}, \qquad \mathcal{P}_0^= = \varnothing.$$

The corresponding gradients are

$$\nabla f^1(\bar{x}) = (1, 0, 0, 0, 0,),$$
$$\nabla f^3(\bar{x}) = (1, 0, 0, \sqrt{2}, \sqrt{2}),$$
$$\nabla f^4(\bar{x}) = (0, -1, 0, 0, 0),$$
$$\nabla f^5(\bar{x}) = (-2, 0, 0, 0, 0).$$

Step 0. Since

$$\nabla f^k(\bar{x}) A_0 = \nabla f^k(\bar{x}) \neq 0, \qquad \text{for all } k \in \mathcal{P}_0,$$

we solve the system given by (10); i.e.,

$$\lambda_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_3 \begin{bmatrix} 1 \\ 0 \\ 0 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix} + \lambda_4 \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_5 \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5 = 1, \qquad \lambda_k \geq 0.$$

A solution is

$$\lambda_1 = 2/3, \qquad \lambda_3 = \lambda_4 = 0, \qquad \lambda_5 = 1/3.$$

Therefore,

$$J_0=\{1,5\}, \qquad \mathscr{P}_1=\{3,4\}, \qquad \mathscr{P}_1^{=}=\{1,5\},$$

$$A_1=\begin{bmatrix}0&0&0\\0&0&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix},$$

$$\mathscr{R}(A_1)=\bigcap_{k\in J_0} D^{=}_{f_0^k P_0}, \qquad P_1=P_0A_1=A_1.$$

Step 1. Since

$$\nabla f^4(\bar{x})P_1=0,$$

while

$$\nabla f^3(\bar{x})P_1\neq 0,$$

we get that

$$J_1=\{4\}, \qquad \mathscr{P}_2=\{3\}, \qquad \mathscr{P}_2^{=}=\{1,4,5\},$$

$$A_2=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},$$

$$\mathscr{R}(A_2)=D^{=}_{f_0^4 P_1}, \qquad P_2=P_1A_2=A_1.$$

Step 2. Since

$$\mathscr{P}_2=\{3\} \quad \text{and} \quad \nabla f^3(\bar{x})P_2\neq 0,$$

Case (iii) occurs. Stop. We conclude that

$$\mathscr{P}^{=}=\mathscr{P}_2^{=}=\{1,4,5\},$$

$$D^{=}_{\mathscr{P}^{=}}=\mathscr{R}(P_2)=\left\{\begin{pmatrix}0\\0\\0\\0\\d_3\\d_4\\d_5\end{pmatrix}\in R^5: d_3, d_4, d_5\in R\right\}.$$

Using the substitution technique, and checking whether

$$\nabla f^k(\bar{x})P_i = 0,$$

reduces the number of computations required to find $\mathcal{P}^=$ and $D^{\bar{=}}_{\mathcal{P}^=}$, compared to the algorithms in Refs. 5 and 6. Another improvement is obtained by using the result in Ref. 8. There it is shown that one need only find

$$A_{i+1} \in R^{n_i \times n_{i+1}},$$

such that

$$\mathcal{R}(A_{i+1}) = D^{\bar{=}}_h, \qquad h = \sum_{k \in \mathcal{P}_i} \lambda_k f^k 0 P_i,$$

rather than having $\mathcal{R}(A_{i+1})$ equal to the intersection of the cones of directions of constancy. One then obtains $D^{\bar{=}}_h$, where

$$h = \sum_{k \in \mathcal{P}^=} \alpha_k f^k, \qquad \alpha_k > 0,$$

rather than $D^{\bar{=}}_{\mathcal{P}^=}$. The cone $D^{\bar{=}}_h$ can be used to regularize (P), instead of the cone $D^{\bar{=}}_{\mathcal{P}^=}$.

Finding $\mathcal{P}^=$ and $D^{\bar{=}}_h$ (or $D^{\bar{=}}_{\mathcal{P}^=}$) is also an unstable process, since we must solve the homogeneous system (10), but the rank of the corresponding matrix, with columns $\phi^k P_i$, is unknown. If we assign a numerical rank to this matrix (see Section 3), then we can solve (10) within a given tolerance $\varepsilon > 0$. Thus, (10) becomes

$$\sum_{k \in \mathcal{P}_i} \lambda_k \phi^k p_i = \phi, \qquad \|\phi\| < \varepsilon, \tag{25}$$

where

$$\lambda_k \geq 0, \qquad \sum \lambda_k = 1, \qquad \phi^k \in \partial f^k(\bar{x}).$$

For simplicity, let us consider step $i = 0$, with

$$P_0 = A_0 = I.$$

Let

$$h_\epsilon = \sum_{k \in J_0} \lambda_k f^k.$$

Note that we can set $\lambda_k = 1$, if J_0 is found using (8) with

$$\phi \in \partial f^k(\bar{x}), \qquad \|\phi\| < \epsilon$$

replacing

$$0 \in \partial f^k(\bar{x}).$$

The ϵ subscript for h denotes that we have solved (25) and found J_0 within a tolerance of $\epsilon \geq 0$. By continuity of the f^k,

$$\phi \in \partial h_\epsilon(\bar{x}).$$

Therefore,

$$h(y) \geq h(\bar{x}) + \phi(y - \bar{x}) \geq -\|\phi\| \, \|y - \bar{x}\|.$$

Thus, in a bounded neighborhood Ω of $\bar{x}$, we can claim that

$$h(y) \geq -\delta, \qquad \text{for all } y \in \Omega, \text{ for some } \delta > 0.$$

This is done by setting the tolerance ϵ sufficiently small. Continuing this reasoning, we see that, rather than finding $\mathcal{P}^{=}$, we find

$$\mathcal{P}^{=}(\epsilon) \supset \mathcal{P}^{=}.$$

Thus, rather than finding $D^{=}_{h}$, where

$$h = \sum_{k \in \mathcal{P}^{=}} \alpha_k f^k, \qquad \alpha_k > 0,$$

we find

$$D^{=}_{h_\epsilon} \subset D^{=}_{h},$$

with the dimensions

$$\dim D^{=}_{h} \leq \dim D^{=}_{h}.$$

This is assuming that we find the cones of directions of constancy exactly. However, in any case, the instability in finding $\mathcal{P}^{=}$ has the opposite effect of the instability in finding the cones of directions of constancy; i.e., we are likely to have

$$k \in \mathcal{P}^{=}(\epsilon) \backslash \mathcal{P}^{=},$$

which results in $D^{=}_{h_\epsilon}$ being smaller than $D^{=}_{h}$. Recall that the algorithm in Section 2 might result in $D^{=}_{h}$ being too large. However, combining the two arguments, we can conclude that, in a bounded neighborhood of $\bar{x}$,

$$|f(\bar{x}) - f(y)| < \epsilon,$$

for a given tolerance $\epsilon > 0$. Moreover, if we can calculate the cones of directions of constancy exactly, perhaps analytically, then

$$D^{=}_{h_\epsilon} \subset D^{=}_{h}.$$

Thus, we get an inner approximation to $D^{=}_{h}$. Or it may happen that we can calculate the dimension of the cones of constancy exactly. Then, we still obtain

$$\dim D^{=}_{h_\epsilon} \leq \dim D^{=}_{h},$$

though the two subspaces $D^{=}_{h_\epsilon}$ and $D^{=}_{h}$ might only be *close* and not equal.

We can now assign a numerical rank to our program (P). We first define the *rank* of (P) to be $\dim D^{=}_{\mathcal{P}^=}$. This is equal to the dimension of the feasible set S.

Definition 4.1. The program (P) has numerical rank (δ, ϵ, r), with respect to the vector norm $\|\cdot\|$, if $r = \inf\{\text{rank } (P_y)$: (P_y) is the perturbed program obtained by perturbing the right-hand sides of the constraints by $y = (y_k)\}$, i.e., $f^k(x) \leq y_k$, with $\|y\| < \epsilon$, and the feasible set $S_y \neq 0$.

$$\epsilon < \delta \leq \sup\{\eta : \|y\| < \eta, S_y \neq \varnothing \Rightarrow \text{rank}(P_y) \geq r\}.$$

Note that it has been shown in Ref. 16 that Slater's condition holds for (P) if and only if the interior of the feasible set, int S, is nonempty. This is due to the fact that we assume f^k, $k \in \mathcal{P}^=$, faithfully convex. Thus, Slater's condition is equivalent to full rank of (P) and to full numerical rank for sufficiently small ϵ. Note also that we might still have a stable program without full rank. For example, a linear program is always stable, but might not satisfy Slater's condition.

5. Method of Reduction

We now collect the machinery presented in the previous two sections and formulate the method of reduction. This algorithm first finds a feasible point and then solves the general convex program (P) when the constraints f^k, $k \in \mathcal{P}^=$, are faithfully convex. Let us denote by $S(P)$ any method that solves program (P) under the assumption that Slater's condition is satisfied (e.g., Zoutendijk's feasible directions method, ref. 2). The method of reduction finds the regularized program (P_r) of Theorem 2.1, in the process of finding a feasible point. It then solves (P_r), using $S(P_r)$. Furthermore, if Slater's condition is not satisfied for the original program (P), the regularized program (P_r) will always have fewer constraints and fewer variables than (P).

Algorithm *C*

Initialization. Let

$$P_0 = I_{n\times n}, \qquad T_0 = \mathcal{P}, \qquad \hat{\mathcal{R}}_0 = \varnothing, \qquad \mathcal{R}_0 = \varnothing, \qquad n_0 = n, \qquad \bar{x}^1 \in R^n.$$

ith step, $1 \le i \le$ card $\mathscr{P}$. Set

$$\mathscr{R}_i = \hat{\mathscr{R}}_{i-1} \cup \{k \in T_{i-1}: f^k(\bar{x}^i) \le 0\}, \tag{26}$$

$$T_i = T_{i-1}/\mathscr{R}_i. \tag{27}$$

Now, consider the program

$$(\mathrm{R_i}) \qquad \sum_{k \in T_i} f^k(\bar{x}^i + P_{i-1}y) \to \min,$$

$$\text{s.t. } f^k(\bar{x}^i + P_{i-1}y) \le 0, \qquad k \in \mathscr{R}_i, \qquad y \in R^{n_{i-1}}.$$

Using the feasible point 0 and Theorem 2.3, regularize this program; i.e., find $\mathscr{R}_i^=$ [note that $\mathscr{R}_i^=$ is the equality set for ($\mathrm{R_i}$)] and the $n_{i-1} \times n_i$ matrix A_i satisfying

$$D^=_{\mathscr{R}_i^=} = \mathscr{R}(A_i)$$

(use Algorithm B). Now, set

$$\hat{\mathscr{R}}_i = \mathscr{R}_i \backslash \mathscr{R}_i^=, \tag{28}$$

$$P_i = P_{i-1}A_i, \tag{29}$$

to get the *reduced program*

$$(\hat{\mathrm{R}}_\mathrm{i}) \qquad \sum_{k \in T_i} f^k(\bar{x}^i + P_i y) \to \min,$$

such that

$$f^k(\bar{x}^i + P_i y) \le 0, \qquad k \in \hat{\mathscr{R}}_i, \qquad y \in R^{n_i}.$$

Note that Slater's condition is satisfied and 0 is a feasible point, by Theorem 2.3.

Case (i). Suppose that

$$T_i = \varnothing.$$

Set

$$\bar{x} = \bar{x}^i \quad \text{and} \quad P = P_i,$$

and solve the reduced program,

$$(\mathrm{R}) \qquad f^\circ(\bar{x} + Py) \to \min,$$

such that

$$f^k(\bar{x} + Py) \le 0, \qquad k \in \hat{\mathscr{R}}_i, \qquad y \in R^{n_i},$$

using the initial feasible point $y = 0$ and $S(\boldsymbol{R})$.

We conclude that, if y^* is a solution of $\boldsymbol{R}$, then

$$x^* = \bar{x} + Py^*$$

is a solution of the original program (P).

Case (ii). Suppose that

$$T_i \neq \varnothing.$$

Then, set $z_0^i = 0$ and

$$z_{j+1}^i = S(\hat{R}_i) z_j^i, \qquad j = 0, 1, \ldots;$$

i.e., z_{j+1}^i is the point obtained after one iteration of $S(\hat{R}_i)$, applied to point z_j^i.

Case (ii) (a). Suppose that, after j iterations of $S(\hat{R}_i)$, we find $k \in T_i$ such that

$$f^k(\bar{x}^i + P_i z_j^i) \leq 0.$$

Then, set

$$z^i = z_j^i, \qquad \bar{x}^{i+1} = \bar{x}^i + P_i z^i, \tag{30}$$

and proceed to step $i+1$.

Case (ii) (b). Suppose that, after j iterations of $S(\hat{R}_i)$, we have not found $k \in T_i$ such that

$$f^k(\bar{x}^i + P_i z_j^i) \leq 0,$$

but z_j^i solves the program ($\hat{\mathrm{R}}_i$).

We conclude that $S = \varnothing$.

Intuitively, the algorithm is straightforward. It uses the usual phase I process of placing the constraints in the objective function and then minimizing and eliminating constraints from the objective function, when they satisfy the feasibility requirements. The proof is given in Ref. 17. A computer program and examples are also presented in Ref. 17.

6. Conclusions

In this paper, we have presented an algorithm that solves the convex program (P) when the constraints f^k, $k \in \mathscr{P}^=$, are faithfully convex. Note that the class of faithfully convex functions includes all analytic as well as all strictly convex functions. The algorithm may also be applied to functions which are piecewise faithfully convex, i.e., to functions which can be written as the supremum of a finite number of faithfully convex functions.

We have not assumed differentiability. For applications of nondifferentiable functions in optimization, see, e.g., Refs. 20, 21, 22, 23, 24. Moreover, no constraint qualification, such as Slater's condition, was required. Situations where Slater's condition, or for that matter any other constraint qualification, fails, arise, for example, in multicriteria problems (Ref. 21). Zoutendijk (Ref. 2) suggests that, when Slater's condition fails, then one should solve the perturbed problem

$$(\mathrm{P}_\epsilon) \qquad \text{minimize } f^\circ(x),$$
$$\text{subject to } f^k(x) \le \epsilon, \qquad k \in \mathscr{P},$$

where $\epsilon > 0$. This may lead to very small step sizes when using feasible direction methods and may cause zigzagging. Note that zigzagging is usually overcome by solving the perturbed problem (P_ϵ) with $\epsilon < 0$. The algorithm presented here suggests the reverse. That is when Slater's condition is satisfied, but the feasible set is thin, then one should use less accuracy, so that Slater's condition fails, i.e., so that

$$\mathscr{P}^=(\varepsilon) \ne \varnothing.$$

This would be equivalent to moving on the boundary or parallel to the boundary of the feasible set. Note that, if $\mu(\epsilon)$ is the optimal value of (P_ϵ), then $\mu(\epsilon)$ converges to the solution of the dual program of (P) as $\epsilon \downarrow 0$. Moreover, if Slater's condition fails, then the Kuhn–Tucker conditions might also fail. In this case, the program (P) is an unstable program, in the sense that

$$\lim_{\varepsilon \downarrow 0} [\mu(\varepsilon) - \mu(0)]/\varepsilon = -\infty$$

(see Ref. 11). Thus, even if there is no duality gap, we obtain an infinite marginal improvement in the optimal value with respect to small positive perturbations.

Now, suppose that Slater's condition is satisfied for (P), but that the feasible set is very thin; i.e., our algorithm finds

$$\mathscr{P}^=(\epsilon) \ne \varnothing,$$

for our chosen tolerance ϵ. This is equivalent to a small (negative) perturbation in the constraints f^k, $k \in \mathscr{P}^=(\epsilon)$, which results in these constraints being identically 0 on the feasible set S. But, since Slater's condition is satisfied, the program (P) is stable with respect to small perturbations. Thus, the algorithm is stable when treating such problems. Note that, if one perturbs (P) to obtain (P_ε) above, then one obtains a thin feasible set.

The regularized program (P_r) satisfies Slater's condition and thus is a stable program. Thus, when solving (P_r), we can obtain valid error estimates using the Kuhn–Tucker vectors. Indeed, (P_r) is equivalent to the program

$$\text{minimize } f^\circ(x),$$
$$\text{subject to } f^k(x) \le 0, \quad k \in \mathscr{P} \backslash \mathscr{P}^=, \quad x \in \bar{x} + D^=_{\mathscr{P}^=},$$

where $\bar{x}$ is any feasible point of (P). Then, if λ is a Kuhn–Tucker vector for (P_r), and $\mu_r(\epsilon)$ denotes the optimal value of the perturbed program of (P_r), with perturbation vector ε, we obtain (see Ref. 11)

$$\mu_r(\varepsilon) - \mu(0) \geq -(\lambda, \varepsilon).$$

This shows that if, while finding $\mathcal{P}^=(\varepsilon)$, we find the cones of directions of constancy of the right dimension, and so

$$\mathcal{P}^=(\varepsilon) \supset \mathcal{P}^=,$$

then the regularized program which we find is a stable perturbation of the original program. Note that finding $\mathcal{P}^=(\varepsilon)$ too large is a stable perturbation, while finding it too small results in instability.

Though the solution that we find may not be stable, due to the instability of calculating the cones of directions of constancy, we can talk of stability, in some sense, by using the numerical ranks. That is, the solution that we find may not be stable with respect to the solution of the original program (P). However, by finding a gap between ϵ and δ [see the definition of the numerical rank of (P)], the solution which we actually find is stable with respect to small perturbation. Thus, the estimate that we find for the solution of (P) is stable with respect to small perturbations. This is the motivation behind the numerical rank in Ref. 19.

To summarize, the algorithm presented here solves, theoretically, convex programs without assuming any constraint qualifications. In practice, due to roundoff errors, it appears to solve the perturbed program (P_ϵ), by using an approximation to the thin feasible set F_ε. It appears particularly well suited for solving problems with thin feasible sets. Instability might arise if Slater's condition fails and the cones of directions of constancy are calculated too large. This appears to be counterbalanced by having $\mathcal{P}^=(\varepsilon)$ large. Several open questions have been raised by the introduction of numerical rank for (P) and remain to be studied. The algorithm has been tested on several problems successfully. The success might be attributed to the fact that the instability in finding $\mathcal{P}^=$ counterbalances, to some extent, the instability in finding $D_k^=$. Moreover, the instability in finding $\mathcal{P}^=$ too large does not result in instability of the algorithm. The damaging instability arises when calculating $D_k^=$. However, Algorithm A has been tested on numerous faithfully convex functions and only failed on specially designed *bad* examples.

References

1. MANGASARIAN, O. L., *Nonlinear Programming*, McGraw-Hill, New York, New York, 1969.

2. ZOUTENDIJK, G., *Methods of Feasible Directions, A Study in Linear and Nonlinear Programming*, Elsevier, New York, New York, 1960.
3. BEN-TAL, A., BEN-ISRAEL, A., and ZLOBEC, S., *Characterization of Optimality in Convex Programming without a Constraint Qualification*, Journal of Optimization Theory and Applications, Vol. 20, pp. 417–437, 1976.
4. BEN-TAL, A., and ZLOBEC, S., *A New Class of Feasible Direction Methods*, University of Texas, Austin, Texas, Research Report No. CCS216, 1977.
5. ABRAMS, R. A., and KERZNER, L., *A Simplified Test for Optimality*, Journal of Optimization Theory and Applications, Vol. 25, pp. 161–170, 1978.
6. BEN-ISRAEL, A., BEN-TAL, A., and ZLOBEC, S., *Optimality in Nonlinear Programming: A Feasible Directions Approach*, John Wiley, New York, New York, 1981.
7. BEN-TAL., and BEN-ISRAEL, A., *Characterizations of Optimality in Convex Programming: The Nondifferentiable Case*, Applicable Analysis, Vol. 9, pp. 137–156, 1979.
8. WOLKOWICZ, H., *A Strengthened Test for Optimality*, Journal of Optimization Theory and Applications, Vol. 35, pp. 397–415, 1981.
9. ABRAMS, R. A., and WU, C. T., *Projection and Restriction Methods in Geometric Programming and Related Problems*, Journal of Optimization Theory and Applications, Vol. 26, pp. 59–76, 1978.
10. ROCKAFELLAR, R. T., *Some Convex Programs Whose Duals Are Linearly Constrained*, Nonlinear Programming, Edited by J. B. Rosen, O. L. Mangasarian, and K. Ritter, Academic Press, New York, New York, pp. 293–322, 1970.
11. ROCKAFELLAR, R. T., *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.
12. GOULD, F. J., and TOLLE, J. W., *Geometry of Optimality Conditions and Constraint Qualifications*, Mathematical Programming, Vol. 2, pp. 1–18, 1972.
13. HOLMES, R. B., *Geometric Functional Analysis and Its Applications*, Springer-Verlag, New York, New York, 1975.
14. DUBOVITSKII, A. YA., and MILYUTIN, A. A., *Extremum Problems in the Presence of Restrictions*, Zurnal Vycislitel'noi Matematiki i Matematiceskoi Fiziki, Vol. 5, pp. 395–453, 1965.
15. ZLOBEC, S., *Mathematical Programming*, McGill University, Lecture Notes, 1977.
16. WOLKOWICZ, H., *Geometry of Optimality Conditions and Constraint Qualifications: The Convex Case*, Mathematical Programming, Vol. 19, pp. 32–60, 1980.
17. WOLKOWICZ, H., *Constructive Approaches to Approximate Solutions of Operator Equations and Convex Programming*, McGill University, PhD Thesis, 1978.
18. WOLKOWICZ, H., *Calculating the Cone of Directions of Constancy*, Journal of Optimization Theory and Applications, Vol. 25, pp. 451–457, 1978.
19. GOLUB, G., KLEMA, V., and STEWART, G. W., *Rank Degeneracy and Least Squares Problems*, Stanford University, Research Report, 1976.

20. BEN-TAL, A., KERZNER, L., and ZLOBEC S., *Optimality Conditions for Convex Semi-Infinite Programming Problems*, Naval Research Logistics Quarterly, Vol. 27, pp. 413–435, 1980.
21. BEN-TAL, A., and ZLOBEC, S., *Convex Programming and the Lexicographic Multicriteria Problem*, Mathematische Operationsforschung und Statistik Optimization, Vol. 8, pp. 61–73, 1977.
22. CLARKE, F. H., *Nondifferentiable Functions in Optimization*, Applied Mathematics Notes, Vol. 2, pp. 84–92, 1976.
23. DEM'YANOV, V. F., and MALOZEMOV, V. N., *Introduction to Minimax*, John Wiley and Sons, New York, New York, 1974.
24. PSHENICHNYI, B. N., *Necessary Conditions for an Extremum*, Marcel Dekker, New York, New York, 1971.