Technical Research Report

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

Fast Feasible Direction Methods, with Engineering Applications

by A.L. Tits and J. Zhou

TR 91-20

European Control Conference
Grenoble, France, July 1991

Fast Feasible Direction Methods, With Engineering Applications*

A.L. Tits and J. Zhou
Department of Electrical Engineering and
Systems Research Center
University of Maryland, College Park, MD 20742

1 Numerical Optimization in Engineering

Optimization problems arising in engineering applications often present distinctive features that are not exploited, or not accounted for, in standard numerical optimization algorithms and software codes.

First, in many cases, equality constraints are not present, or can be simply eliminated (e.g., by an appropriate change of variables, or, say, by explicit integration of systems dynamics). Thus promising approaches that do not have a provision for handling equality constraints should not be discarded.

Second, there are several instances where it is advantageous, or even crucial, that, once a feasible point (point satisfying all constraints) has been achieved, all the subsequent iterates generated by the algorithm be feasible as well. One such instance is in real time, closed loop optimal control applications. There, an acceptable control action (i.e., satisfying all "hard" constraints) must be determined within a specific amount of time; applying a control that satisfies certain constraints may also be mandated for stability of the closed loop system to be ensured (e.g., [1]). Other reasons why it may be useful to generate feasible iterates include (i) the possibility that the objective function be not well defined outside the feasible set (e.g., a time domain L_2 error for an unstable dynamical systems) and (ii) in a multiobjective problem, the impracticability of meaningfully exploring tradeoffs (interactively), when hard constraint violations make many options unacceptable. Thus there is a clear need for efficient algorithms generating feasible iterates.

Third, as just suggested, many optimization problems arising in engineering are best formulated as (constrained) multiobjective problems. A natural way to tackle such problems is to express them as (constrained) weighted minimax problems, with weights interactively adjusted to explore tradeoffs or steer the solution to a "true" optimal (e.g., [2,3]). Constrained minimax problems should thus be given special attention.

Fourth, especially in dynamical systems design or control applications, some specifications must be achieved over a range of values of an independent parameter (time, frequency, etc.). This gives rise to "functional" objectives or constraints, resulting in "semi-infinite" optimization problems. Again, one needs ways to efficiently tackle such problems.

While various other distinctive features arise in optimization problems found in specific classes of engineering problems, this paper focuses on those identified above, as they have been the object of special attention by the authors and their co-workers in recent years. Thus, throughout, it is dealt only with optimization problems that do not involve nonlinear equality constraints. Sections 2 to 5 below discuss (i) a simple scheme that generates a sequence of feasible iterates converging superlinearly, (ii) an improvement of this scheme via introduction of a nonconventional line search rule, (iii) an extension that efficiently handles minimax problems, and (iv) another extension, to semi-infinite programming problems. An underlying assumption throughout is that, as is often the case in engineering problems, function evaluations are computationally demanding, much more so than, e.g., solution of a quadratic program.

Due to space constraints, no convergence analysis is included; it will appear elsewhere.

2 Basic FSQP Scheme

Feasible direction methods have been proposed and analyzed by several authors as far back as in the early 1960's (e.g., [4–6]) and extensions to semi-infinite problems have been devised (e.g., [7,8]). While classical feasible direction methods have been used successfully on many problems, they suffer from a major drawback: because they are "first order" methods, i.e., they do not make use of some kind of estimate of second derivatives, convergence is typically slow. Recently, it was shown that the popular, superlinearly convergent, Sequential Quadratic Programming (SQP) iteration could be adapted so as to produce a sequence of feasible iterates [9,10]. A simple

*This research was supported in part by NSF's Engineering Research Centers Program No. NSFD-CDR-88-03012, by NSF grant No. DMC-88-15996 and by a grant from the Westinghouse Corporation.

version is presented now.

Consider the problem

$$(P_1) \quad \text{minimize} \quad f(x) \quad \text{s.t.} \quad x \in X$$

where

$$X = \{x \text{ s.t. } g(x) \le 0\},$$

with $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ smooth functions. Note that, when all iterates satisfy the constraints, it is natural that the line search (necessary to enforce global convergence) be such that the objective function f decrease at each iteration. Thus we intend to enforce the conditions

$$x_{k+1} \in X \tag{2.1}$$

$$f(x_{k+1}) \le f(x_k) \tag{2.2}$$

at each iteration k. Let $x_k \in X$ be an estimate of a strong minimizer x^* for (P_1) and let H_k be a symmetric positive definite approximation to the Hessian of the Lagrangian function. The traditional SQP iteration consists of first computing a search direction d_k^0 by solving the quadratic program $QP_1(x_k, H_k)$

$$\begin{array}{ll} \min\limits_{d^0} & \frac{1}{2}\langle d^0, H_k d^0\rangle + \langle \nabla f(x_k), d^0\rangle \\ \text{s.t.} & g_j(x_k) + \langle \nabla g_j(x_k), d^0\rangle \le 0, \quad \forall j. \end{array} \tag{2.3}$$

It is readily checked that d_k^0 is a descent direction for f at x_k. However, d_k^0 may not be feasible at x_k since (2.3) allows $\langle \nabla g_j(x_k), d_k^0\rangle = 0$ for active constraints. Also, even when d_k^0 is feasible, a line search enforcing (2.1) and (2.2) may not allow a full step of one to be taken in a neighborhood of a solution and thus superlinear convergence may never take place. It is shown in [10] that these difficulties can be circumvented by means of two simple mechanisms corresponding to successively "tilting" and "bending" the search direction.

Specifically, first d_k^0 is replaced by a convex combination $d_k = (1-\rho_k)d_k^0 + \rho_k d_k^1$ of d_k^0 and some (essentially arbitrary) feasible descent direction d_k^1, e.g., d_k^1 is the minimizer for the quadratic program

$$\begin{array}{ll} \min\limits_{d^1,\delta} & \langle d_k^0 - d^1, d_k^0 - d^1\rangle + \delta \\ \text{s.t.} & \langle \nabla f(x_k), d^1\rangle \le \delta \\ & g_j(x_k) + \langle \nabla g_j(x_k), d^1\rangle \le \delta, \ \forall j. \end{array}$$

To ensure that d_k inherits the quasi-Newton character of d_k^0, ρ_k is forced to go to 0, e.g.,[1]

$$\rho_k = \frac{\|d_k^0\|^2}{1 + \|d_k^0\|^2}.$$

Second, d_k is "bent", i.e., a search along an arc $x_k + t d_k + t^2 \tilde{d}_k$ is performed. This is because, even close to x^*, $x_k + d_k$ may violate both feasibility and decrease of f. Such an arc search was previously used by Mayne and Polak [11] in a different context. Here, $\tilde{d}_k$ is selected in such a way that, (i) in a neighborhood of x^*, $x_{k+1} (= x_k + d_k + \tilde{d}_k)$ satisfies both (2.1) and (2.2), (ii) $d_k + \tilde{d}_k$ converges to d_k, so as to preserve the quasi-Newton character of the iteration. This requires that, upon approaching a solution to (P_1), $x_k + d_k + \tilde{d}_k$ closely follow the active constraint boundaries from inside the feasible set, which can be achieved, e.g., by performing a Newton-like iteration at $x_k + d_k$. Specifically, to take advantage of the available second order information H_k, $\tilde{d}_k$ is defined to be the solution of the quadratic program $\widetilde{QP}_1(x_k, d_k, H_k)$

$$\begin{array}{ll} \min\limits_{\tilde{d}} & \frac{1}{2}\langle (d_k + \tilde{d}), H_k(d_k + \tilde{d})\rangle + \langle \nabla f(x_k), \tilde{d}\rangle \\ \text{s.t.} & g_j(x_k + d_k) + \langle \nabla g_j(x_k), \tilde{d}\rangle \le -\|d_k\|^\gamma, \quad \forall j \end{array} \tag{2.4}$$

(but $\tilde{d}_k$ is discarded if too large).[2] Here, $\gamma \in (2,3)$ is preselected; the range of values (2,3) insures that the correction is significant enough to restore the feasibility, yet $x_k + d_k + \tilde{d}_k$ is close enough to the boundaries of the active constraints to ensure descent on f. This yields the following algorithm, dubbed FSQP (Feasible Sequential Quadratic Programming).

Algorithm 1

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $\gamma \in (2,3)$.

Data. $x_0 \in X$, $H_0 \in \mathbb{R}^{n \times n}$, and $H_0 = H_0^T > 0$.

Step 0. Initialization. Set $k = 0$.

Step 1. Computation of a search arc.

i. Compute d_k^0, the solution of $QP_1(x_k, H_k)$. If $d_k^0 = 0$, stop.

ii. Select suitable d_k^1 and ρ_k and set

$$d_k = (1-\rho_k)d_k^0 + \rho_k d_k^1.$$

iii. Compute $\tilde{d}_k$, the solution of $\widetilde{QP}_1(x_k, d_k, H_k)$. If $\|\tilde{d}_k\| > \|d_k\|$, set $\tilde{d}_k = 0$.

Step 2. Arc search. Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying

$$f(x_k + t d_k + t^2 \tilde{d}_k) \le f(x_k) + \alpha t \langle \nabla f(x_k), d_k\rangle$$

$$g_j(x_k + t d_k + t^2 \tilde{d}_k) \le 0, \quad \forall j.$$

Step 3. Updates. Compute a new approximation H_{k+1} to the Hessian of the Lagrangian. Set $x_{k+1} = x_k + t_k d_k + t_k^2 \tilde{d}_k$. Set $k = k+1$. Go back to Step 1. □

Global and local convergence of Algorithm 1 are established in [10] under mild assumptions.

[1] $\rho_k = O(\|d_k^0\|^2)$ is necessary to make it possible for the subsequent correction $\tilde{d}_k$ to ensure (2.2).

[2] Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$.

3 Nonmonotone Line Search

Computation of the correction $\tilde{d}_k$ in Step 1 iii of Algorithm 1 involves an additional evaluation of the constraints at each iteration, at point $x_k + d_k$ (see (2.4)). It turns out that, if (2.2) is not required, x_{k+1} may follow less closely the boundaries of the active constraints without jeopardizing superlinear convergence; evaluation of $g(x_k + d_k)$ can then be dispensed with and replaced by a certain "trial and error" boundary search carried from iteration to iteration (see below). The key issue is then to insure global convergence without enforcing (2.2), i.e., without monotone decrease of the objective function. It turns out that this can be achieved, by suitably adapting a "nonmonotone line search" scheme introduced by Grippo, Lampariello and Lucidi in the context of Newton's method for unconstrained optimization [12]. A detailed exposition of such extension, with complete analysis, can be found in [13] and [14]. Here we recall the essentials of this approach.

First it turns out that, if a sequence of iterates $\{x_k\}$ is generated by the basic SQP iteration for (P_1), i.e., $x_{k+1} = x_k + d_k^0$ where d_k^0 solves the quadratic program $QP_1(x_k, H_k)$, if x_0 is sufficiently close to a strong local minimizer x^* of (P_1) and if the entire sequence $\{x_k\}$ happens to be feasible, then, given any positive number α,

$$f(x_{k+1}) \leq f(x_{k-3}) + \alpha\langle\nabla f(x_k), x_{k+1} - x_k\rangle \quad (3.1)$$

will hold for k large enough. Thus, a line search rule requiring that the condition

$$f(x_k + td_k^0) \leq \max_{\ell=0,\cdots,3} f(x_{k-\ell}) + \alpha t\langle\nabla f(x_k), d_k^0\rangle$$

be satisfied would eventually *always* accept the full step of one, and superlinear convergence would be preserved.[3]

However, as in §2, d_k^0 need not be feasible. To ensure the feasibility of the full step of one close to a solution, consider a search direction d_k such that

$$d_k = d_k^0 + O(\|d_k^0\|^2) \quad (3.2)$$

satisfying

$$g_j(x_k) + \langle\nabla g_j(x_k), d_k\rangle \leq -C\|d_k^0\|^2, \quad \forall j, \quad (3.3)$$

for given $C > 0$. An exponent less than 2 on the right hand side of (3.3) (i.e., a larger correction) would make (3.2) impossible to satisfy and, as a consequence, (3.1) may not hold with the iteration $x_{k+1} = x_k + d_k$. If (3.3) holds, the sequence $\{x_k\}$ constructed by the iteration $x_{k+1} = x_k + d_k$ will satisfy

$$\begin{aligned} g_j(x_{k+1}) &= g_j(x_k) + \langle\nabla g_j(x_k), d_k\rangle \\ &\quad + \frac{1}{2}\langle d_k, \frac{\partial^2 g_j}{\partial x^2}(\bar{x}_{k+1})d_k\rangle \\ &\leq -C\|d_k^0\|^2 + \frac{1}{2}\|d_k\|^2\|\frac{\partial^2 g_j}{\partial x^2}(\bar{x}_{k+1})\|, \forall j \end{aligned}$$

with $\bar{x}_{k+1} = x_k + \xi_{j,k}d_k$ for some $\xi_{j,k} \in [0,1]$. And if (3.2) holds, for k large enough we will have $g_j(x_{k+1}) \leq 0$, $\forall j$, provided $2C$ is strictly larger than the largest among all eigenvalues of the Hessians $\frac{\partial^2 g_j}{\partial x^2}(x^*)$, $\forall j$. While these Hessians are obviously unknown, it is shown in [14] that one can adaptively obtain a suitably large value of C, by increasing C whenever $x_k + d_k$ is not feasible.

As in §2, d_k will be chosen as a convex combination of d_k^0 and a certain direction d_k^1. To ensure that (3.3) can be achieved close to x^*, it will be required that d_k^1 be a (nonzero) direction of strict feasibility even at stationary points. Clearly then, one cannot any more require that d_k^1 be a direction of descent for f. For example, d_k^1 can be chosen to be the minimizer for the quadratic program

$$\begin{aligned} \min_{d^1,\delta} \quad & \|d^1\|^2 + \delta \\ \text{s.t.} \quad & g_j(x_k) + \langle\nabla g_j(x_k), d^1\rangle \leq \delta, \ \forall j. \end{aligned}$$

The following algorithm is devised to implement these ideas.

Algorithm 2

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $\underline{C}$, $\underline{d} > 0$.

Data. $x_0 \in X$, $H_0 \in \mathbb{R}^{n\times n}$, and $H_0 = H_0^T > 0$.

Step 0. Initialization. Set $k = 0$, $x_{-3} = x_{-2} = x_{-1} = x_0$, and $C_0 = \underline{C}$.

Step 1. Computation of a new iterate.

i. Compute d_k^0, the solution of $QP_1(x_k, H_k)$. If $d_k^0 = 0$, stop.

ii. Select d_k^1 with the required properties. Let $v_k = \min\{C_k\|d_k^0\|^2, \|d_k^0\|\}$ and let $\rho_{k,j}$ be the lesser number between 1 and the smallest $\rho \geq 0$ such that

$$g_j(x_k) + \langle\nabla g_j(x_k), (1-\rho)d_k^0 + \rho d_k^1\rangle \leq -v_k\,.$$

Finally, let $\rho_k = \max_j \{\rho_{k,j}\}$.

iii. Obtain a search direction

$$d_k = (1-\rho_k)d_k^0 + \rho_k d_k^1.$$

iv. Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying

$$f(x_k + td_k) \leq \max_{\ell=0,\cdots,3}\{f(x_{k-\ell})\} + \alpha t\langle\nabla f(x_k), d_k\rangle$$

$$g_j(x_k + td_k) \leq 0, \quad \forall j$$

and set $x_{k+1} = x_k + t_k d_k$.

Step 2. Updates. (i) If $\|d_k^0\| > \underline{d}$, set $C_k = \underline{C}$. Otherwise, if $g_j(x_k + d_k) \leq 0$, $\forall j$, set $C_{k+1} = C_k$. Otherwise, set $C_{k+1} = 2C_k$. (ii) Compute a new approximation H_{k+1} to the Hessian of the Lagrangian. Set $k = k + 1$. Go back to Step 1.

□

[3] The "max" insures that the line search will complete successfully, even away from x^*.

Algorithm 2 as just stated may not meet our requirements. Specifically, (3.1) can only be shown to hold provided a full step is taken at each iteration, in particular, at iterations $k-3$, $k-2$ and $k-1$. Thus the algorithm must be suitably *initialized*. In [14], it is shown that this can be done by making use (if necessary) in the early iterations of a bending-and-arc-search technique such as that discussed in §2 above. Also, in the early iterations, d_k may not be a descent direction for f, and it may be necessary to use a different direction, closer to d_k^0. Under mild assumptions, global and local superlinear convergence can be shown to hold [14].

4 Minimax Problems

Now consider the "minimax" problem

$$(P_2) \quad \text{minimize} \quad f(x) \quad \text{s.t.} \quad x \in X$$

where $f(x) = \max_{i \in \{1,\cdots,p\}} f_i(x)$ with $f_i : \mathbb{R}^n \to \mathbb{R}$ smooth functions. For the simplicity of exposition, we first assume that $X = \mathbb{R}^n$. Problem (P_2) can be readily solved by Algorithm 2, using a standard transformation. However, it is shown in [15] that one can use a simpler scheme to solve (P_2) more efficiently. It is also shown that, for an SQP direction d_k^0 (see, e.g., [16,17] for the use of an SQP scheme in the context of minimax problems), the solution of the quadratic program $QP_2(x_k, H_k)$

$$\min_{d^0} \ \frac{1}{2}\langle d^0, H_k d^0 \rangle + f'(x_k, d^0)$$

with

$$f'(x,d) = \max_i \{f_i(x) + \langle \nabla f_i(x), d \rangle\} - f(x)$$

a first order approximation to $f(x+d) - f(x)$ at x in direction d, given any positive number α, the three-step monotone (rather than four-step monotone, cf. (3.1)) condition[4]

$$f(x_k + d_k^0) \le f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle \qquad (4.1)$$

eventually holds, locally around a strong minimizer x^* for (P_2). Here again, the algorithm must be initialized, e.g., by a bending-and-arc-search scheme. A suitable correction $\tilde{d}_k$, to be computed whenever (4.1) does not hold, can in the present case be obtained by solving the quadratic program $\widetilde{QP}_2(x_k, d_k^0, H_k)$

$$\min_{\tilde{d}} \ \frac{1}{2}\langle (d_k + \tilde{d}), H_k(d_k + \tilde{d}) \rangle + \tilde{f}'(x_k + d_k^0, x_k, \tilde{d})$$

(but $\tilde{d}_k$ is discarded if too large), with

$$\tilde{f}'(x+d, x, \tilde{d}) = \max_i \{f_i(x+d) + \langle \nabla f_i(x), \tilde{d} \rangle\} - f(x+d).$$

[4] $\langle d_k^0, H_k d_k^0 \rangle$, a lower bound to $|f'(x_k, d_k^0)|$, can also be used in the line searches of Algorithms 1 and 2, but with apparently no significant advantage.

This suggests the following algorithm.

Algorithm 3

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$.

Data. $x_0 \in X$, $H_0 \in \mathbb{R}^{n \times n}$, and $H_0 = H_0^T > 0$.

Step 0. Initialization. Set $k = 0$ and $x_{-2} = x_{-1} = x_0$.

Step 1. Computation of a new iterate.

i. Compute d_k^0, the solution of $QP_2(x_k, H_k)$. If $\|d_k^0\| = 0$, stop.

ii. If

$$f(x_k + d_k^0) \le \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha \langle d_k^0, H_k d_k^0 \rangle,$$

set $t_k = 1$, $\tilde{d}_k = 0$ and go to Step 2.

iii. Compute $\tilde{d}_k$, the solution of $\widetilde{QP}_2(x_k, d_k^0, H_k)$. If $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

iv. Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying

$$f(x_k + t d_k^0 + t^2 \tilde{d}_k) \le \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha t \langle d_k^0, H_k d_k^0 \rangle.$$

Step 2. Updates. Compute a new approximation H_{k+1} to the Hessian of the Lagrangian. Set $x_{k+1} = x_k + t_k d_k^0 + t_k^2 \tilde{d}_k$. Set $k = k+1$. Go back to Step 1. □

If constraints are present, a suitable combination of Algorithm 2 and Algorithm 3 will do. Again, global and local superlinear convergence follow under mild assumptions [15].

5 Functional Specifications

Of interest here are problems such as (P_1) but for which X (in the simplest case) is of the form

$$X = \{x \text{ s.t. } \psi(x,\omega) \le 0 \quad \forall \omega \in [0,1]\}$$

where $\psi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is smooth. To make the exposition as simple as possible, however, we consider instead the closely related problem

$$(P_3) \quad \text{minimize} \quad \max_{\omega \in [0,1]} \phi(x,\omega)$$

where $\phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is smooth. The difficulties in solving (P_3) stem from the fact that (i) accurate evaluation of $\max_{\omega \in [0,1]} \phi(x,\omega)$ requires a potentially time consuming global maximization and (ii) $\max_{\omega \in [0,1]} \phi(x,\omega)$ is typically nonsmooth. Much of the existing literature focuses on the local solution of (P_3) (e.g., papers in [18]). Globally convergent algorithms typically rely on an adaptively refined *discretization* of the interval of variation of ω (e.g., [7, 8]). In this context, a key issue is how to efficiently solve the discretized problem

$$(P_3') \quad \text{minimize} \quad \max_{\omega \in \Omega} \phi(x,\omega)$$

with

$$\Omega = \{0, \frac{1}{q}, \frac{2}{q}, \cdots, \frac{q-1}{q}, 1\}$$

where q is a fixed positive integer. While this problem is of the form (P_2), regularity of ϕ as a function of ω ought to be exploited (especially when q is large). In [7] and [8], (P_3') is solved by means of first order (i.e., slow) methods. In [7] the search direction at iteration k is based on the gradients $\nabla_x \phi(x_k, \omega)$ at all "ε-maximizing" values of $\omega \in \Omega$. In [8], it is shown that a smaller set of gradients can be used by suitably detecting "critical" values ω and "remembering" them from iteration to iteration.

Here we propose to efficiently solve (P_3') by adapting the ideas developed in [8] to the framework of Algorithm 3 above. $QP_2(x_k, H_k)$ and $\widetilde{QP}_2(x_k, d_k^0, H_k)$ will be replaced by $QP_3(x_k, H_k, \Omega_k)$ and $\widetilde{QP}_3(x_k, d_k^0, H_k, \Omega_k)$. The latter are identical to the former except that $f_i(\cdot)$ and $\nabla f_i(\cdot)$ are replaced by $\phi(\cdot, \omega)$ and $\nabla\phi(\cdot, \omega)$ and the maximizations are now over Ω_k, an adaptively constructed subset of Ω. Specifically, Ω_{k+1} includes all ω's that globally maximize $\phi(x_{k+1}, \cdot)$ over Ω, all ω's that affected direction d_k^0, as well as all ω's that restricted the step length at iteration k. Based on these ideas, the following algorithm is proposed. For any $x \in \mathbb{R}^n$, we define

$$f(x) = \max_{\omega \in \Omega} \phi(x, \omega)$$

and the set of global maximizers at x

$$\hat{\Omega}(x) = \{\omega \in \Omega : \phi(x, \omega) = f(x)\}.$$

Algorithm 4

Parameters. $\alpha \in (0, \frac{1}{2}), \beta, \in (0, 1)$.

Data. $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$, and $H_0 = H_0^T > 0$.

Step 0. Initialization. Set $k = 0$, $x_{-2} = x_{-1} = x_0$, and $\Omega_0 = \hat{\Omega}(x_0)$.

Step 1. Computation of a new iterate.

i. Compute d_k^0, the solution of $QP_3(x_k, H_k, \Omega_k)$. If $\|d_k^0\| = 0$, stop.

ii. If

$$f(x_k + d_k^0) \le \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha\langle d_k^0, H_k d_k^0\rangle,$$

set $t_k = 1$, $\tilde{d}_k = 0$ and go to Step 2.

iii. Compute $\tilde{d}_k$, the solution of $\widetilde{QP}_3(x_k, d_k^0, H_k, \Omega_k)$. If $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

iv. Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying

$$f(x_k + t d_k^0 + t^2 \tilde{d}_k) \le \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha t\langle d_k^0, H_k d_k^0\rangle.$$

Step 2. Updates. Set $x_{k+1} = x_k + t_k d_k^0 + t_k^2 \tilde{d}_k$. Define

$$\Omega_{k+1} = \hat{\Omega}(x_{k+1}) \cup \{\omega_i \in \Omega_k : \lambda_{k,i} > 0\}$$
$$\cup\{\omega \in \Omega : f(\bar{x}_{k+1}) > f(x_k) - \alpha\bar{t}_k\langle d_k^0, H_k d_k^0\rangle\}$$

where $\bar{x}_{k+1} = x_k + \bar{t}_k d_k^0 + \bar{t}_k^2 \tilde{d}_k$ with $\bar{t}_k = \min\{1, \frac{t_k}{\beta}\}$ and $\lambda_{k,i}$ is the multiplier associated with each element $\omega_i \in \Omega_k$ in solving $QP_3(x_k, H_k, \Omega_k)$. Compute a new approximation H_{k+1} to the Hessian of the Lagrangian. Set $k = k + 1$. Go back to Step 1.

□

Here also, global and local superlinear convergence are established under mild assumptions [19].

6 Implementation and Applications

The algorithms presented in this paper are at different stages of implementation. The FSQP Fortran code [20] encompasses Algorithms 1 through 3, with simple extension to efficiently handle affine constraints, including affine equality constraints. Updating of H_k is by means of the BFGS formula with Powell's modification [21]. FSQP has been extensively tested on standard test problems [10,14,15]. It appears to be competitive with the currently most popular algorithms (not of feasible direction type). Algorithm 2 is noticeably superior to Algorithm 1. The FSQP code has also been used on engineering applications. Examples include predictive control using neural net dynamic models [22] and batch process optimal profile determination [23] (both of them are real time applications). A "distributed computation" implementation is being completed and will soon be tested on civil engineering structural design problems [24].

Algorithms 1 through 4 (except, for the time being, for the nonmonotone line search) have been implemented in the CONSOLE package [25], an interactive system for optimization-based design of engineering systems. Success has already been achieved on a number of problems, including the design of mechanical components [26] and that of a multivariable controller for an advanced rotorcraft [27].

References

[1] D.Q. Mayne & H. Michalska, "Receding Horizon Control of Nonlinear Systems," *IEEE Trans. Automat. Control* AC-35 (1990), 814–824.

[2] W.T. Nye & A.L. Tits, "An Application-Oriented, Optimization-Based Methodology for Interactive Design of Engineering Systems," *Internat. J. Control* 43 (1986), 1693–1721.

[3] A.L. Tits, M.K.H. Fan & E.R. Panier, "Aspects of Optimization-Based CADCS," *4th IFAC Symposium on Computer-Aided Design in Control Systems*, Beijing, China (August 23-25, 1988).

[4] G. Zoutendijk, *Methods of Feasible Directions*, Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 1960.

[5] S.I. Zukhovitskii, R.A. Polyak & M.E. Primak, "An Algorithm for the Solution of Convex Programming Problems," *DAN USSR* 153 (1963), 991–1000.

[6] E. Polak, *Computational Methods in Optimization*, Academic Press, New York, N.Y., 1971.

[7] E. Polak & D.Q. Mayne, "An Algorithm for Optimization Problems with Functional Inequality Constraints," *IEEE Trans. Automat. Control* 21 (1976), 184–193.

[8] E.R. Panier & A.L. Tits, "A Globally Convergent Algorithm with Adaptively Refined Discretization for Semi-Infinite Optimization Problems Arising in Engineering Design," *IEEE Trans. Automat. Control* AC-34 (1989).

[9] E.R. Panier & A.L. Tits, "A Superlinearly Convergent Feasible Method for the Solution of Inequality Constrained Optimization Problems," *SIAM J. Control Optim.* 25 (1987), 934–950.

[10] E.R. Panier & A.L. Tits, "On Feasibility, Descent and Superlinear Convergence in Inequality Constrained Optimization," Systems Research Center, University of Maryland, Technical Report SRC-TR-89-27, College Park, MD 20742, 1989.

[11] D. Q. Mayne & E. Polak, "A Superlinearly Convergent Algorithm for Constrained Optimization Problems," *Math. Programming Stud.* 16 (1982), 45–61.

[12] L. Grippo, F. Lampariello & S. Lucidi, "A Nonmonotone Line Search Technique for Newton's Method," *SIAM J. Numer. Anal.* 23 (1986), 707–716.

[13] E.R. Panier & A.L. Tits, "Avoiding the Maratos Effect by Means of a Nonmonotone Line Search. I. General Constrained Problems," *SIAM J. Numer. Anal.* 28 (August 1991, to appear).

[14] J.F. Bonnans, E.R. Panier, A.L. Tits & J. Zhou, "Avoiding the Maratos Effect by Means of a Nonmonotone Line Search. II. Inequality Constrained Problems – Feasible Iterates," Systems Research Center, University of Maryland, Technical Report SRC-TR-89-42r1, College Park, MD 20742, 1989.

[15] J. Zhou & A.L. Tits, "Nonmonotone Line Search for Minimax Problems," 1991, in preparation.

[16] S.P. Han, "Variable Metric Methods for Minimizing a Class of Nondifferentiable Functions," *Math. Prog.* 20 (1981), 1–13.

[17] E.R. Panier & A.L. Tits, "A Superlinearly Convergent Method of Feasible Directions for Optimization Problems Arising in the Design of Engineering Systems," in *Proc. of the Seventh International Conference on Analysis and Optimization of Systems — Antibes, June 25-27, 1986*, A. Bensoussan & J.L. Lions, eds., Lecture Notes in Control and Information Sciences #83, Springer Verlag, Berlin-Heidelberg-New York-Tokyo, June 1986, 65–73.

[18] R. Hettich, *Semi-Infinite Programming*, Lecture Notes in Control and Information Sciences #15, Springer Verlag, 1979.

[19] J. Zhou & A.L. Tits, 1991, in preparation.

[20] J. Zhou & A.L. Tits, "User's Guide for FSQP Version 2.0 – A Fortran Code for Solving Optimization Problems, Possibly Minimax, with General Inequality Constraints and Linear Equality Constraints, Generating Feasible Iterates," Systems Research Center, University of Maryland, SRC TR-90-60r1, College Park, MD 20742, 1990.

[21] M.J.D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," in *Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630*, G.A. Watson, ed., Springer-Verlag, 1978, 144–157.

[22] J.S. Donat, N. Bhat, T.J. McAvoy & J. Zhou, "Predictive Control Using Neural Net Dynamic Models and Optimization," *IMACS Congress*, Dublin, Ireland (July, 1991).

[23] E. Zafiriou & J.M. Zhu, "Iterative Batch-to-Batch Optimal Profile Determination for Batch and Semi-Batch Processes," Systems Research Center, University of Maryland, College Park, MD 20742, 1991, in preparation.

[24] B.K. Voon & M.A. Austin, "Structural Optimization in a Distributed Computing Environment," Systems Research Center, University of Maryland, College Park, MD 20742, 1991, in preparation.

[25] M.K.H. Fan, L.-S. Wang, J. Koninckx & A.L. Tits, "Software Package for Optimization-Based Design with User-Supplied Simulators," *IEEE Control System Magazine* 9 (1989).

[26] S. Azarm, A.L. Tits & M.K.H. Fan, "Optimization-Based Design of Mechanical Systems Using CONSOLE," *submitted for possible presentation at the NATO Advanced Study Institute*, Berchtesgaden, Germany (September 23-October 4, 1991).

[27] M.K.H. Fan, A.L. Tits, J. Barlow, N.K. Tsing, M. Tischler & M. Takahashi, "On the Design of Decoupling Controllers for Advanced Rotorcraft in the Hover Case," paper AIAA-91-0421, presented at the AIAA 29th Aerospace Sciences Meeting, Reno, Nevada, January 7-10, 1991.