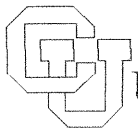
A Practical Class of Globally Convergent Active Set Strategies for Linearly Constrained Optimization *

Richard H. Byrd
Gerald A. Schultz

CU-CS-238-82



University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

* This research was supported by NSF grant MCS 81-15475.

A PRACTICAL CLASS OF GLOBALLY CONVERGENT ACTIVE SET STRATEGIES FOR LINEARLY CONSTRAINED OPTIMIZATION

by

Richard H. Byrd and Gerald A. Shultz

CU-CS-238-82 September, 1982

Department of Computer Science University of Colorado at Boulder, Boulder, Colorado 80309

This research was supported by NSF grant MCS 81-15475.

ABSTRACT

We present here conditions on active set strategies for linearly constrained optimization which guarantee global convergence and non-zigzagging. The conditions amount to requiring that no constraint be dropped immediately after a new constraint has been hit, and if a constraint has a negative multiplier at a stationary point, that a constraint be dropped if the iterate is close enough. It is shown that a number of practical algorithms, some very close to those used in practice, satisfy these conditions. For global convergence, the only conditions on the problem are continuity of first derivatives and non-degeneracy. In addition we give a partial extension of a global convergence result of Fletcher regarding a constraint dropping strategy recently proposed by him.

1. Introduction

Many widely used algorithms for minimization of a nonlinear objective function subject to linear constraints make use of active set strategies. That is, a certain set of the constraints are treated as equality constraints and the resulting subproblem can be handled by a reduced version of any iterative unconstrained minimization algorithm on the resulting subspace. Such algorithms require that at times a constraint is added to the active set of constraints and at times is removed from the active set. This paper is concerned with rules for making these decisions.

There have been a large number of criteria proposed for deciding when to drop a constraint. They range from requiring exact minimization on a subspace to dropping a constraint whenever its Lagrange multiplier has the wrong sign. There would seem to be some advantage to allowing constraints to be dropped before a subspace minimum has been found since steps taken toward the minimum on the wrong subspace are probably not making progress toward the solution. Computational experiments by Lenard[7] on a number of such strategies ranging between these two extremes seem to indicate that strategies which are freer to drop constraints tend to require fewer iterations.

It is well known, however, that an algorithm which can drop constraints on every step is open to the possibility of zigzagging, that is, oscillation among several active sets without settling down on the correct one. This is illustrated in an example due to Wolfe[12] where such an algorithm zigzags and converges to a point which does not satisfy the Kuhn-Tucker conditions. This means, of course, that it is impossible to prove global convergence for an algorithm with such a strategy. It also suggests that even if this kind of false convergence did not occur much effort might be wasted in switching

subspaces before the right one is found. Even in the neighborhood of a minimizer zigzagging can slow convergence, as demonstrated in an example of Zoutendijk[13].

In part motivated by this situation, a number of strategies lying between the two extremes have been proposed. A reasonable ad hoc rule is to estimate the decrease in the quadratic model of the objective which would result without dropping and with dropping, and to drop if the extra decrease due to dropping is sufficient. This is suggested by a number of researchers [5,10,11] and is a natural criterion, but it gives no guarantee of global convergence. Zoutendijk[13] suggests that a constraint be dropped at first if a multiplier estimate has the wrong sign, but if the constraint is added again it should not be dropped until a subspace minimum is reached. This guarantees global convergence, but is very restrictive and may result in solving many equality constrained problems. A number of other strategies are discussed in a survey by Fletcher[1]. One may also weaken the tolerance for convergence on a subspace, making it easier to drop constraints, but again at the loss of any guarantees of proper convergence on the whole problem. A dropping rule recently proposed by Fletcher[2] could be regarded as a subspace stopping rule designed to avoid zigzagging. He proves a global convergence result for it in the case of a strictly convex objective function. It is probably one of the least restrictive dropping rules proposed that has any theoretical guarantees. Later in this paper we will give a partial extension of his result to the non-convex case.

It should be made clear that the active set strategies we are discussing here work by considering an equality constrained subproblem at each iteration. There are other strategies which take into consideration constraints which are not currently active in determining the next step. One could, for example, minimize a quadratic model of the objective subject to all the

constraints of the whole problem, just as is done for nonlinearly constrained problems. An algorithm along these lines was proposed by Garcia-Palomares[3] and proved globally convergent; this is also a special case of the algorithm analyzed by Han[6] for general constraints, and shown to be globally convergent. The bending technique of McCormick[8,9] also falls into this general category since the step can be partly determined by the inactive inequality constraints. Indeed, in some cases it can even solve the same subproblem as the successive quadratic programming technique. Global convergence has also been proved by McCormick for his technique. In this paper, however, we will be concerned with methods that consider only the active constraints in determining the step direction.

The main objective of this paper is to suggest a new condition on dropping rules which is unrestrictive and can be shown to guarantee global convergence for any continuously differentiable objective function. The essential condition is that no constraint be dropped from an active set if the previous step involved adding a constraint. The global convergence result is valid for any dropping rule satisfying this condition. The advantage of the condition is that it is easy to modify an ad hoc rule to make it globally convergent without changing its essential character. Although zigzagging is not a great problem in practice, we believe that theory should to some extent account for that fact, and that the lack of provably globally convergent algorithms for linearly constrained optimization is a hindrance to advancement in this area.

In Section 2 we describe the condition and carry out the convergence analysis, and in Section 3 we discuss the relation of our rule with other rules.

2. A Practical Class of Constraint Dropping Rules

In this section we will describe a class of practical rules for dropping constraints which result in a globally convergent algorithm. To do this we

give a general framework for an active set strategy algorithm to solve the problem

$$\min_{A^T x \leq b} f(x): R^n \to R$$

where A is an n by m constraint matrix.

Call A_k the matrix of columns of A that are to be kept active at x_k, and let Z_k be a matrix whose columns are an orthonormal basis for the null space of A_k.

Algorithm 1.

0) Given x_1, pick A_1 to be a matrix whose columns are a linearly independent subset of the constraints satisfied with equality at x_1. Set $k=1$.

1) Compute Z_k, and $d_k \in range(Z_k)$.

2) Compute Lagrange multiplier estimate λ_k. If the dropping rule so indicates, then drop a constraint a_i with $\lambda_k^{(i)}<0$ and update Z_k and d_k as in 1).

3) Do a backtracking linesearch using d_k to get a scalar α_k, and augment A_k if a constraint is hit by the step $\alpha_k d_k$. Set $x_{k+1}=x_k+\alpha_k d_k$.

4) If the stopping conditions are met, then stop.

5) Set $k=k+1$, and go to 1).

Assumptions:

1. For a fixed $\mu>0$, $\nabla f(x_k)^T d_k \leq -\mu \, || Z_k{}^T \nabla f(x_k) || \, || d_k ||$.

2. For any subsequence x_{k_j}, $Z_{k_j}^T g_{k_j} \to 0$ if and only if $d_{k_j} \to 0$.

3. The multipliers λ_k are chosen so that if $x_{k_j} \to x_*$ with $\nabla f(x_*) \in range(A_{k_j})$ for all j, then $\lambda_{k_j} \to \lambda_*$, where constraints which are not active at a point are assumed to have zero multipliers.

4. For a fixed $\eta>0$, whenever a constraint is dropped, the ratio of its multiplier to the most negative multiplier must be greater than η.

5. The scalar α_k is the first scalar in the sequence $\beta_1 \geq \beta_2 \geq \cdots$ satisfying

$$f(x_k) - f(x_k + \beta_j d_k) \geq \gamma \beta_j \nabla f(x_k)^T d_k,$$

where $\beta_1 = 1$ and each $\beta_{j+1} \geq \rho \beta_j$ for a fixed $\rho > 0$, and $\gamma \varepsilon (0,1)$.

When we refer to Algorithm 1 the listed assumptions will be considered as holding, but the dropping rule is left unspecified at this point.

The above algorithm is intended to be a general framework and to allow for various implementations. In many specific algorithm implementations the direction d_k will be the solution to

$$\min_{A_k^T w = 0} \tfrac{1}{2} w^T H_k w + \nabla f(x_k)^T w \quad , \tag{2.1}$$

which has the form $-Z_k (Z_k^T H_k Z_k)^{-1} Z_k{}^T \nabla f(x_k)$, if $Z_k^T H_k Z_k$ is positive definite. If $Z_k^T H_k Z_k$ and its inverse are uniformly bounded in norm, then this d_k satisfies assumptions 1 and 2.

A good value of λ is given by the solution to

$$H_k d_k + \nabla f(x_k) + A_k \lambda = 0,$$

which is the exact Lagrange multiplier for problem (2.1). This method for computing the multipliers satisfies the conditions of the algorithm. It also guarantees that if d_k is always generated by a quadratic model as described above and if a constraint is dropped, then the modified direction is feasible with respect to the dropped constraint as is shown by Gill and Murray[4]. Given that λ_k is not non-negative, and that the dropping rule is satisfied, a constraint with negative multiplier must be picked to be dropped, satisfying

assumption 4 of Algorithm 1. One choice would be to pick the most negative $\lambda_k^{[i]}$. Another would be to drop the constraint which would yield the greatest reduction of the quadratic model if it were dropped and no constraint were hit. This would also satisfy assumption 4 as long as $||H_k||$ remains bounded for all k.

The linesearch procedure specified in assumption 5 covers the case of backtracking by a constant factor or backtracking by interpolation as long as the ratio of successive linesearch iterates is bounded. Our analysis could also apply to other linesearches such as the first local minimizer or Goldstein-Armijo conditions if they are modified so that α_k is bounded uniformly.

We now consider the issue of deciding when to drop a constraint at all. By "dropping rule" in step 2) of the algorithm, we mean a test which, if satisfied and if the multipliers are not all nonnegative, indicates that a constraint be dropped. We give here a condition on the constraint dropping rule which guarantees global convergence, but which allows much latitude in the precise nature of the rule.

Full Step Dropping Rule Condition

> The dropping rule must be such that a constraint is dropped only if there is a negative multiplier and no constraint was added on the last step, and such that a constraint is always dropped if the above holds with x_k sufficiently close to a stationary point with a negative Lagrange multiplier.

Many possible dropping rules satisfy this condition. For example a relatively unconstrained rule would result if a constraint were dropped whenever the "only if" part of the condition is satisfied. One might also want to impose some subsidiary conditions as long as they are not stronger than the "if" part

of the condition. Some such rules are discussed in Section 3.

An advantage of our basic condition, in addition to the convergence properties to be demonstrated, is that the need for using Lagrange multiplier estimates based on the same quadratic model used in generating the direction is less critical. If, based on some other multiplier estimates, a direction infeasible with respect to the dropped constraint is generated, that constraint may be immediately readded and a null step taken, possibly causing cycling. The condition of not dropping immediately after an add would prevent this particular form of cycling.

Some global convergence results about Algorithm 1, with various dropping rules, will now be proved. The following definition and lemmas help to clarify the proofs of the theorems to come.

Definition When discussing a cluster point x_* of a sequence generated by Algorithm 1, we will call a set of constraints and a corresponding matrix $\hat{A}$ deficient if there is no solution λ to $\nabla f(x_*)+\hat{A}\lambda=0$.

Note that deficiency of a constraint set implies that $\hat{Z}^T\nabla f(x_*)\neq 0$, where $\hat{Z}$ is a null space matrix for $\hat{A}$.

Lemma 1 Suppose x_* is a cluster point of Algorithm 1. Then for any $\varepsilon>0$ there is an $r>0$ and an integer K such that if $k>K$ and $||x_k-x_*||<r$ then $||\alpha_k d_k||<\varepsilon$.

Proof. For $||x_k-x_*||$ sufficiently small, if A_k is non-deficient then $Z_k{}^T\nabla f(x_k)$ is arbitrarily small, and thus by assumption 2 of Algorithm 1, $||d_k||$ is arbitrarily small. Since $\alpha_k\leq 1$, the result holds for non-deficient A_k.

Since there are only finitely many Z_k and ∇f is continuous, there is a $\delta>0$ and an $r>0$ such that for all k with A_k deficient and $||x_k-x_*||<r$,

$||Z_k{}^T\nabla f(x_k)||\geq\delta$. So, for all k with $||x_k-x_*||<r$ and A_k deficient,

$$f(x_k)-f(x_k+\alpha_k d_k)\geq-\gamma\alpha_k\nabla f(x_k)^T d_k$$
$$\geq\gamma\alpha_k||Z_k{}^T\nabla f(x_k)||\ ||d_k||\geq\gamma\delta||\alpha_k d_k||\ .$$

Hence, with K such that $f_k-f_{k+1}\leq\frac{\varepsilon}{\gamma\delta}$ for $k\geq K$, the result follows.

Lemma 2 Suppose x_* is a cluster point of Algorithm 1. Then there is an $r>0$ and an integer K such that if $k>K$, $||x_k-x_*||<r$, and A_k is deficient, then a new constraint must be added at x_{k+1}.

Proof. Call $\delta=\min\{\frac{Z^T\nabla f(x_*)}{2}: Z$ corresponds to a deficient constraint set $A\}$. Since there are only finitely many constraint sets, $\delta>0$, and for any x_k close enough to x_*, with A_k deficient, $||Z_k{}^T\nabla f(x_k)||>\delta$, by the continuity of ∇f.

Now, note that there can not be infinitely many x_k with A_k deficient and $\alpha_k=1$, by assumption 2 of Algorithm 1, since $||Z_k{}^T\nabla f(x_k)||>\delta$, but by Lemma 1, $\alpha_k d_k=1d_k$ becomes arbitrarily small.

Thus, beyond some index, if the step to x_{k+1} does not hit a new constraint, by assumption 5 of Algorithm 1,

$$f(x_k)-f(x_k+\xi_k d_k)\leq-\gamma\xi_k\nabla f(x_k)^T d_k\ ,$$

where $\alpha_k\geq\xi_k\rho$ for fixed $\rho>0$. Then for some $\beta_k\varepsilon(0,\xi_k)$,

$$f(x_k)-f(x_k+\xi_k d_k)=-\xi_k\nabla f(x_k+\beta_k d_k)^T d_k\ .$$

So, subtracting $\nabla f(x_k)^T d_k$ from both sides of the above inequality and using assumption 1 of Algorithm 1,

$$(\nabla f(x_k+\beta_k d_k)-\nabla f(x_k))^T d_k\geq(\gamma-1)\nabla f(x_k)^T d_k\geq(1-\gamma)\mu||Z_k{}^T\nabla f(x_k)||d_k||\ .$$

Hence for all x_k close enough to x_* with deficient active set,

$$||\nabla f(x_k+\beta_k d_k)-\nabla f(x_k)||\geq(1-\gamma)\mu\,||Z_k{}^T\nabla f(x_k)||\geq(1-\gamma)\mu\delta>0\,.$$

But $\beta_k\leq\xi_k\leq\frac{\alpha_k}{\rho}$, so by Lemma 1 $||\alpha_k d_k||$, and hence $||\beta_k d_k||$, becomes arbitrarily small; the above inequality contradicts the continuity of ∇f.

The main global convergence result now follows easily from the above lemmas and the full step dropping rule condition.

Theorem 1 Suppose $f:R^n\rightarrow R$, $f\,\varepsilon C^1(R^n)$, and x_* is a cluster point of the iterates generated by Algorithm 1 using a dropping rule satisfying the full step dropping rule condition. Then if the active set at x_* is linearly independent, x_* is a Kuhn-Tucker point.

Proof. First, suppose to the contrary that A_* is deficient. Let A_M be an active set occurring arbitrarily close to x_* with the largest number of constraints among such sets. When we say that A_M occurs at x_k we mean that A_M is the set of constraints to be held active upon taking the step from x_k. Now consider an iterate x_k, with active set A_M, close enough to x_* and with k large enough that Lemma 2 applies. If A_* is deficient, all active sets at iterates close enough to x_* will also be deficient. So, by Lemma 2, the step from x_k must add a new constraint, and by the dropping rule condition no constraint may be dropped upon leaving x_{k+1}, so the active set at x_{k+1} is larger than A_M. This contradicts the choice of A_M, since iterates with the properties of x_k occur arbitrarily close to x_*. Thus, there is a λ_* such that $\nabla f(x_*)+A_*\lambda_*=0$.

Next we will prove that $\lambda_*\geq 0$. Suppose rather that for at least one i, $\lambda_*^{(i)}<0$. Consider A_M as defined above. By Lemma 1, let k be large enough and x_k close enough to x_* that each of x_k, x_{k+1}, x_{k+2}, and x_{k+3} is close enough to x_* that Lemma 2 applies. Also, pick x_k so that $A_k=A_M$. Now, by the choice of A_M there are clearly infinitely many such x_k for which no constraint is added on the step from x_k to x_{k+1}, thus we may pick x_k with $A_{k+1}=A_M$, ini-

tially. Further, note that by Lemma 2, A_M must be non-deficient, since no constraint was added. So, for x_k close enough to x_*, by assumption 3 of Algorithm 1, $\lambda_k^{(i)}$ will be less than zero for some i, thus by the dropping condition a constraint will be dropped. The active set with which the step from x_{k+1} is taken will therefore have one less constraint than A_M, and will consequently, by the linear independence of A_* and assumption 4 of Algorithm 1, be deficient. So, by Lemma 2, a constraint must be added at x_{k+2}. Note that the constraint which is added can not be the one which was just dropped, unless the step d_{k+1} was not feasible with respect to the constraint dropped. But for x_k close enough to x_* that is impossible, since

$$0=(\nabla f(x_*)+A_*\lambda_*)^T \frac{d_{k+1}}{||d_{k+1}||}=\frac{d_{k+1}^T \nabla f(x_*)}{||d_{k+1}||}+\lambda_*^{(i)} \frac{d_{k+1}^T a_i}{||d_{k+1}||}$$

$$\leq -\mu ||Z_{k+1}^T \nabla f(x_*)|| + \lambda_*^{(i)} \frac{d_{k+1}^T a_i}{||d_{k+1}||},$$

and since the first quantity in the last inequality is bounded below zero for deficient constraint sets, and $\lambda_*^{(i)}<0$, $d_{k+1}^T a_i \leq 0$. But, finally, since the original constraint dropped is still not in the active set at x_{k+2}, that active set is still deficient, and so again by Lemma 2, another constraint must be added at x_{k+3}, and by the dropping condition, no constraint may be dropped. This is a contradiction, since we have shown that infinitely many iterates arbitrarily close to x_* have active sets larger than A_M. Thus, $\lambda_* \geq 0$, and x_* is a Kuhn-Tucker point, as was to have been shown.

Note that Theorem 1 only assumes continuity of the derivative of f as opposed to Lipschitz continuity. This degree of generality is worth noting since the well known example of convergence to a non-stationary point due to Wolfe[12] involves an objective function which is not Lipschitz continuous. Although this is somewhat pathological, we feel the example is significant in that a method that exhibited false convergence for an objective function with

an unbounded Hessian might get temporarily bogged down on a similar problem with a Hessian that became very large and ill-conditioned.

False convergence is not the only difficulty associated with zigzagging. When the algorithm is converging to a Kuhn-Tucker point it is also desirable to have it settle down on a single active set instead of constantly switching. To guarantee this it is necessary to additionally assume that the point satisfies strict complementarity and the sufficient conditions for a strict local minimum.

Theorem 2 Suppose that the point x_* is a cluster point of a sequence generated by Algorithm 1, and that x_* is a Kuhn-Tucker point such that $\nabla f(x_*)+A_*\lambda_*=0$ with A_* of full rank, $\lambda_*>0$, all other constraints are strictly satisfied, and $d^T\nabla^2 f(x_*)d>0$ for all $d\neq 0$ such that $A_*^T d=0$. Then the sequence converges to x_* and the active set remains constant for all k sufficiently large.

Proof. The hypotheses imply that x_* is the only Kuhn-Tucker point within some distance r of x_*, and thus the only cluster point within that neighborhood. Therefore, for any $r'<r$ there are only finitely many iterates such that $r'<||x_k-x_*||<r$. But by Lemma 1, for x_k close to x_* and k large enough, $\alpha_k d_k$ is arbitrarily small and so $||x_{k+1}-x_*||<r$. Thus the entire sequence must converge to x_*. Now for k sufficiently large the sequence is close enough to x_* that no constraints outside of A_* are satisfied with equality, and Lemma 2 applies. Then the active set at an iterate x_k is a subset of A_*, and if it is a proper subset it is deficient and so a constraint must be added at the next step. So within m steps $A_k=A_*$, and by assumption 3 and strict complementarity, $\lambda_k>0$, so no constraints are dropped and the active set remains fixed, as was to be shown.

A result of this type is very important because once the iteration settles on one subspace, then one can rely on local convergence properties for equality constrained problems. If the active set is constantly changing such guarantees of convergence rate do not apply.

3. Relation to Other Constraint Dropping Rules

We now want to compare our dropping condition with various proposed dropping rules and indicate how the condition might lead us to modify these rules.

The "least constrained" dropping rule which satisfies our condition is to drop a constraint whenever there is a negative multiplier and no constraint was hit on the previous step. This rule seems to allow dropping of constraints quite freely but still guarantees global convergence and non-zigzagging under the assumptions of Theorems 1 and 2.

In spite of this guarantee it may be more efficient to make more effort on the current subspace rather than drop a constraint. One common suggestion is to impose some degree of accuracy on the subspace minimization and only drop when, say, the projected gradient has norm less than some tolerance. Our theory indicates that if in addition to the subspace minimization tolerance it is required that no constraint was hit on the previous step, the resulting algorithm is globally convergent. This immediately follows from Theorem 1 since the first part of our condition is imposed directly, and near a stationary point the tolerance is eventually satisfied, as required in the second part.

An additional condition, suggested by Gill and Murray[4], is that the first and second order multiplier estimates agree fairly closely. This will happen when sufficiently close to a nondegenerate stationary point, and as in the preceding paragraph global convergence is assured.

An alternative suggested by a number of researchers[5,10,11] is to compute the decrease in the quadratic model of the objective which would result from a step on the current active set and the additional decrease which would result from dropping a given constraint. These quantities are given by

$$-\tfrac{1}{2}\nabla f(x_k)^T Z_k (Z_k{}^T H_k Z_k)^{-1} Z_k{}^T \nabla f(x_k) \tag{3.1}$$

and

$$-\tfrac{1}{2}\frac{(\lambda_k^{(i)})^2}{(a_k^T H_k^{-1} a_k)_{ii}} \tag{3.2}$$

respectively. If the quantity (3.2) is sufficiently large relative to (3.1) then the constraint may be dropped. This is intuitively a very reasonable condition but no convergence results have been been proved for it. However, if we add the condition that no constraint was hit on the last step, the global convergence and non-zigzagging results of Section 2 follow. This follows since as we approach a stationary point on a subspace, the quantity (3.1) goes to zero, but if there is a negative multiplier at the stationary point, (3.2) stays significantly negative.

A rule which has some similarity to our condition has recently been proposed by Fletcher[2]. We consider it here.

Fletcher's Dropping Rule

Drop a constraint if there is a negative multiplier and

$$\Delta_k < f(x_{l(k)}) - f(x_k), \tag{3.3}$$

where

$$\Delta_k = \tfrac{1}{2}\nabla f(x_k)^T Z_k (Z_k{}^T H_k Z_k)^{-1} Z_k{}^T \nabla f(x_k)$$

and $l(k)$ is the last index where a constraint was dropped.

For this rule Fletcher proves in the case of a uniformly convex objective function that cluster points are Kuhn-Tucker points. Using techniques

similar to those in section 2 we can prove the following partial extension of Fletcher's global convergence results.

Theorem 3 Suppose $f:R^n \to R$, $f \varepsilon C^1(R^n)$ and that $\{x_k\}$ is a sequence of iterates generated by Algorithm 1 using the Fletcher dropping rule. Then any cluster point can be guaranteed to satisfy all the Kuhn-Tucker conditions except $\lambda \geq 0$.

If we assume further that the level set $\{ x : A^T x \leq b \ , f(x) \leq f(x_1) \}$ is bounded and that the active set at any cluster point is of full rank, then the sequence has at least one cluster point which satisfies all the Kuhn-Tucker conditions.

Proof. Call the set of constraints held active at a point before any constraint is dropped the preliminary active set, and, as usual, call the set of constraints with which the step is computed the active set.

Note that Lemmas 1 and 2 of course still hold, since they are independent of the dropping rule. Also, the Fletcher dropping rule clearly implies that if x_k is an iterate close enough to a cluster point x_*, with k large enough and preliminary active set A_k deficient, then a constraint may not be dropped at x_k, since Δ_k will be bounded away from 0, while $f_{l(k)} - f_k$ becomes arbitrarily small.

First, suppose that x_* is a cluster point of the sequence but A_* is deficient. Then all active sets near x_* are deficient, and we can not drop a constraint near x_*. Let A_M be an active set of maximum cardinality occurring arbitrarily close to x_*. Then for x_k with active set A_M close enough to x_* with k large enough, Lemma 2 will apply at x_k and by Lemma 1 x_{k+1} will also be close to x_*. Hence, by Lemma 2 the preliminary active set at x_{k+1} will have one more constraint than A_M has, and since x_{k+1} is close to x_*, no constraint may be dropped, thus the active set at x_{k+1} will have one more constraint than A_M has. But this contradicts the choice of A_M, so in fact there

must be a λ_* such that $\nabla f(x_*)+A_*\lambda_*=0$.

Next, note that there are either infinitely many iterates where a constraint is dropped, or else eventually the active set is constant. If beyond some point no constraints are dropped, then $f_{l(k)}-f_k$ is bounded away from zero for all k. But then, if x_* is a cluster point, say with $x_{k_j}\to x_*$, then $\Delta_{k_j}\to 0$, so we must have $\lambda_*\geq 0$, or else by the Fletcher dropping rule, a constraint would be dropped. Thus, if constraints are dropped only finitely many times, then every cluster point satisfies all the Kuhn-Tucker conditions.

Otherwise, suppose that infinitely many constraints are dropped, that the level set is bounded, and that the active set at every cluster point is of full rank. Let A_M be the largest active set occurring infinitely often at iterates where a constraint is dropped. Since the level set is bounded, we can find a point x_* which is a cluster point of iterates where a constraint is dropped with active set A_M. By the first part of the theorem, there is a λ_* with $\nabla f(x_*)+A_*\lambda_*=0$. To show that this x_* satisfies all the Kuhn-Tucker conditions, suppose to the contrary that for some i, $\lambda_*^{(i)}<0$. Consider an iterate x_k where a constraint is dropped close enough to x_* with active set A_M and k large enough that by Lemma 1, x_k and x_{k+1} are close enough to x_* that Lemma 2 applies, and that a constraint can not be dropped if the preliminary active set is deficient. For x_k close enough, since A_* has full rank, by assumption 4 A_M must be deficient, since a constraint was just dropped, so by Lemma 2 the preliminary active set at x_{k+1} will have one more constraint than A_M has. Now, for x_k close enough, since $\lambda_*^{(i)}<0$, as in the proof of Theorem 1 we see that the step from x_k to x_{k+1} will be feasible with respect to the constraint dropped from the preliminary active set at x_k, and so that constraint can not be the one which is added at x_{k+1}. Thus the preliminary active set at x_{k+1} is still deficient. Hence, by the earlier comments, no con-

straint can be dropped at x_{k+1}, that is, the active set at x_{k+1} will have one more constraint than A_M. Since the active set at x_{k+1} is clearly deficient, again by Lemma 2 we have that the preliminary active set at x_{k+2} will have two more constraints than A_M. Thus, for each iterate after x_{k+1}, up to and including the first of these at which a constraint is dropped, the preliminary active set will have at least two more constraints than A_M has. So, at the iterate where the next constraint is dropped, the active set will have at least one more constraint than A_M has, which contradicts the choice of A_M as the largest active set occurring infinitely often at iterates where a constraint is dropped. Therefore, $\lambda_* \geq 0$, and x_* satisfies all the Kuhn-Tucker conditions as was to be proved.

Note that this theorem extends Fletcher's result in that no convexity is required and the first derivative is only required to be continuous. However it is not as strong as Theorem 1 in that we can only guarantee that at least one of the cluster points is a Kuhn-Tucker point, rather than all of them.

If, in addition to the conditions of Theorem 3, we assume that the objective is strictly convex, then the cluster point which is a Kuhn-Tucker point is the unique minimizer and, by monotonicity of the sequence, is the only cluster point. This is a slightly more general version of Fletcher's result.

Fletcher also proposes a variation of his rule which differs only in that $l(k)$ is the decrease in the objective function since the active set last changed. This modified rule clearly satisfies the "only if" part of our dropping rule condition, since if a constraint was added in reaching x_k then $l(k)=k$ so (3.3) cannot be satisfied. It can also be shown to satisfy the whole condition in the case of a uniformly convex objective function.

In summary, we have seen that with a slight modification a variety of active set strategies can be made globally convergent and non-zigzagging. It

seems likely that the requirement of taking a full step with the current active set before dropping would not affect the performance of the algorithm in most cases. Indeed, the fact that so little work is required on each subspace may be an indication of why zigzagging is seldom observed in practice.

4. References

[1] R. Fletcher, "Minimizing general functions subject to linear constraints", in: F.A. Lootsma, ed., **Numerical Methods for Nonlinear Optimization** (Academic Press, London, 1972).

[2] R. Fletcher, **Practical Methods of Optimization, Vol. 2** (John Wiley & Sons, New York, 1981) 113-117.

[3] U.M. Garcia-Palomares, "Superlinearly convergent algorithms for linearly constrained optimization", in: O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., **Nonlinear Programming 2** (Academic Press, London, 1975) 101-119.

[4] P.E. Gill and W. Murray, "The computation of Lagrange-multiplier estimates for constrained minimization", **Math. Prog. 17** (1979) 32-60.

[5] D. Goldfarb, "Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints", **SIAM J. Appl. Math. 17** (1972) 739-764.

[6] S.P. Han, "A globally convergent method for nonlinear programming", **J.O.T.A. 22** (1977) 297-309.

[7] M. Lenard, "A computational study of active set strategies in nonlinear programming with linear constraints", **Math. Prog. 16** (1979) 81-97.

[8] G.P. McCormick, "Anti-zigzagging by bending", **Mgnt. Sci. 15** (1969) 315-320.

[9] G.P. McCormick, "A second order method for the linearly constrained nonlinear programming problem", in: J.B. Rosen, O.L. Mangasarian, and K. Ritter, eds., **Nonlinear Programming** (Academic Press, London and New York, 1970) 207-243.

[10] B.A. Murtagh and R.W.H. Sargent, "A constrained minimization method with quadratic convergence", in: R. Fletcher, ed., **Optimization** (Academic Press, London, 1969).

[11] J.B. Rosen, "The gradient projection method for non-linear programming, Part I: Linear constraints", **J. SIAM 8** (1960) 181-217.

[12] P. Wolfe, "On the convergence of gradient methods under constraint", **IBM J. Res. and Dev. 16** (1972) 407-411.

[13] G. Zoutendijk, "Nonlinear programming, computational methods", in: J. Abadie, ed., **Integer and Nonlinear Programming** (North-Holland Publishing Co., Amsterdam, 1970).