A Convergent Framework for Constrained Nonlinear Optimisation*

Ron Dembo and Siddhartha Sahi

YALEU/DCS/TR-303

**Yale School of Organization and Management
Box 1-A
New Haven, CT 06520

†Department of Mathematics
Yale University
New Haven, CT 06520

School of Organization and Management Working Paper, Series B #69

September 1983, Revised January 1984

*This research supported in part by DOT Grant CT-06-0011 and by NSF Grant ECS-8119513.

An earlier version of this paper was presented at the XI. International Symposium on Mathematical Programming, Bonn, August 1982.

Abstract

We describe a simple, practical algorithmic framework for constrained nonlinear optimization. Any algorithm that may be expressed in this framework, and most existing algorithms can, will converge to a local minimum from an arbitrary feasible starting point. The framework is particularly suitable for the analysis and development of algorithms for large-scale optimization, since it permits radical changes in the active set to occur at each step and can be implemented in terms of quantities that are easily computed.

1. Introduction

Consider the canonical constrained nonlinear programming problem:

$$\textbf{CNLP} \qquad \text{Minimize: } f_0(x) \tag{1}$$

$$\text{Subject to } f(x) = 0 \tag{2}$$

$$l \leq x \leq u \tag{3}$$

where f_0: $R^n \rightarrow R$, f: $R^n \rightarrow R^m$ and $u, l \in R^n$ are given vectors.

A feasible direction method for CNLP is one which starts at a feasible point x_0 and produces a set of feasible directions, p_k, and feasible points, x_k, with $x_{k+1} = x_k + p_k(\alpha_k)$ where α_k is the steplength at iteration k and $p_k(\alpha_k)$ is referred to as the kth step. The vector function p: $R^+ \rightarrow R^n$ is also referred to as a "search direction" and $\alpha \geq 0$ is the stepsize along this direction. The primary reason for defining p as a function of α is because our framework encompasses algorithms for which $p(\alpha)$ is a curved arc along some nonlinear constraint surface as in Generalized Reduced Gradient (GRG) algorithms [17] or $p(\alpha)$ is a piecewise linear/curved arc in the case of projection methods [1, 2]. For the purposes of this paper we require $p(\alpha)$ to be a continuous function of α for $\alpha \geq 0$. A very common situation is the case where $p(\alpha)$ is linear, *i.e.*, $p(\alpha) = \alpha d$ and d is a search direction computed at x.

Since the inception of linear programming over 35 years ago, numerous feasible direction methods have been proposed for CNLP (see, for example, Fletcher [10, 11], Zoutendijk [28] and Polak [22] for details and extensive references). Still, as yet there is no simple, convergent framework for such methods. It is true that many convergence theorems have been published, each of which imposes a variety of conditions, and with the possible exception of a few [3, 11, 16] are all impractical to test in computer implementations, especially for large-scale programming.

Unfortunately, it is difficult to extract the essential ingredients for a convergence theorem by studying the myriad of conditions and theorems that have been proposed for various algorithms in the literature. For instance, nothing as simple or elegant as the Goldstein-Armijo framework for unconstrained problems (see Dennis and Schnabel [9] or Ortega and Rheinboldt [21] for example) exists for CNLP. The purpose of this paper is to take a step towards such a simple framework by decomposing the convergence problem into manageable components. We provide a framework that is sufficiently general to encompass almost all existing descent algorithms for CNLP, ranging from projection to active-set methods. Furthermore, we provide a set of

conditions that are sufficient to guarantee convergence. These conditions help focus attention on the aspects of an algorithm for CNLP that are crucial to convergence. Once one realizes that they need to be satisfied, they guide both the development of algorithms and convergence proofs.

2. Background and Notation

Before describing our framework we need to present some notation.

Definition 2.1 (Active Set, $\mathcal{A}(x)$)

At a given point x, we define the **active set** *$\mathcal{A}(x)$, as the set of equality constraints (2) taken together with all inequality constraints (3) that hold as equations at x.*

In what follows we will need to work with a restriction of CNLP to some given active set. We use $R(\mathcal{A})$ to denote the **restriction** of CNLP with constraints $i \in \mathcal{A}$ treated as equalities and all remaining inequality constraints ignored. The gradient of $f(x)$ is denoted by $g(x)$.

Definition 2.2 (Critical Point)

A critical point of the problem $R(\mathcal{A})$ is one that satisfies first order optimality conditions.

Definition 2.3 (Optimal Point)

An optimal point is a critical point of CNLP.

3. A New, Globally-Convergent Framework

Our framework involves two types of feasible steps:

1. **restricted** steps, which are defined as feasible descent steps restricted to lie in a manifold containing the current active set, and
2. **relaxing** steps, which are defined as feasible descent steps along which one or more constraints may be relaxed.

Definition 3.1 (Gradient-related steps)

A sequence of **relaxing** *descent steps $p_k(\alpha)$, at points $\{x_k\}$ is said to be* **gradient-related** *if along any convergent subsequence $\{x_{k_i}\} \to \bar{x}$, with $\bar{x}$ not a critical point of CNLP, we have*

$$\lim_{\{x_{k_i}\} \to \bar{x}} \quad [g(x_{k_i})]^\top p_{k_i} \; < \; 0 \; .$$

A sequence of **restricted** *descent steps* $\{p_k^{\mathcal{A}}(\alpha)\}$, *computed at points* $\{x_k\}$, *is said to be* **gradient-related** *if, along any convergent subsequence* $\{x_{k_i}\} \to \bar{x}$, *of points restricted to the active set* $\mathcal{A}$, *with* $\bar{x}$ *not a critical point for the restricted problem* $R(\mathcal{A})$, *we have*

$$\lim_{x_{k_i} \to \bar{x}} \quad [g(x_{k_i})]^T p_{k_i}^{\mathcal{A}} < 0$$

where $p^{\mathcal{A}}$ *is a direction that is feasible and lies on the manifold generated by the constraints in* $\mathcal{A}$.

One way of obtaining gradient-related **relaxing** steps is to project the negative gradients onto the feasible region (see, for example, Bertsekas [1], Goldstein [15], Levitin and Polyak [18]). Another way is to move-off a single constraint with a negative multiplier estimate (see, for example, Fletcher [11] and Gill and Murray [13]).

To compute gradient related **restricted** steps one may use, for example, a restriction of the negative gradient to the active set as is done in Rosen's projected gradient method [24] or Wolfe's reduced gradient method [13] or the variable reduction method of McCormick [20].

Essentially, the motivation for defining relaxing and restricted steps is that any method that generates a sequence of feasible points uses some combination of such steps.

Definition 3.2 (Acceptable points)

Let $p(\alpha)$ *be a descent step computed at a feasible point,* x. *We refer to a point* $x^+ = x + p(\alpha)$ *as* **acceptable** *if* x^+ *is feasible and the step* $p(\alpha)$, $\alpha \geq 0$ *satisfies* **either**

(a) both Goldstein-Armijo conditions

(GA1): $f(x^+) \leq f(x) + \gamma g(x)^T p(\alpha)$; $\gamma \in (0, 1)$

(GA2): $g(x^+)^T p(\alpha) \geq \beta g(x)^T p(\alpha)$; $\beta \in (\gamma, 1)$

or

(b) $\alpha = \bar{\alpha}$ *and* x^+ *satisfies GA1 only; where* $\bar{\alpha} > 0$ *is the maximum distance along* $p(\alpha)$ *before any additional constraints become active.*

Remark 3.1

For the purposes of our main global convergence theorem, GA2 may be replaced by any of the standard conditions (see Fletcher [11], for example) that bound the stepsize away from zero at noncritical points. Both GA1 and GA2 may also be replaced by an Armijo rule [21] or modified Armijo rule [1] if a backtracking linesearch is used.

We are now in a position to describe our framework.

Algorithmic Framework #1

START with x_0 feasible

(Major Iteration; Index $= k$**)**

IF optimal at x_k **THEN** exit.

(Minor Iteration)

ELSE compute a **relaxing** step $p_k(\alpha_k)$

and an **acceptable point** $x_k^+ = x_k + p_k(\alpha_k)$,

set $y \leftarrow x_k^+$;

WHILE a **constraint relaxation condition** is not satisfied at y,

compute a **restricted** step $p(\alpha)$

and an **acceptable point** $y^+ = y + p(\alpha)$,

set $y = y^+$ and **repeat.**

ELSE $k \leftarrow k + 1$

$x_k = y$

start a new major iteration.

Schematically, one may represent a major iteration as is shown in Figure 3.1 . That is, an acceptable relaxation step followed (possibly) by a sequence of acceptable restricted steps.

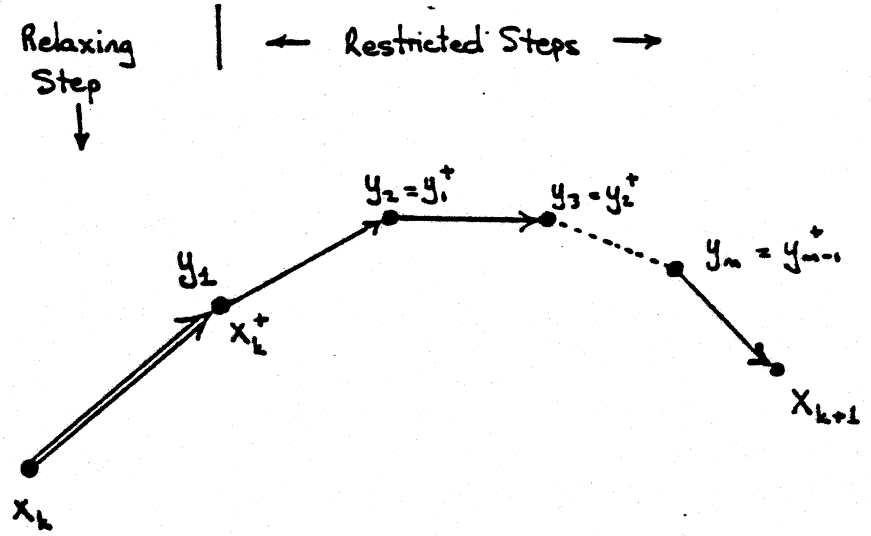

Figure 3.1 A Major Iteration

Remark 3.2

There is no limit to the number of constraints that may be dropped or added respectively, in any single step. For example, each step might involve a projection in which many constraints could be added (in the case of restricted steps) or dropped (in the case of relaxing steps). A description of how this may be achieved is given in Section 5.

The only aspect of this framework that has been left (deliberately) vague is the "constraint relaxation condition". This is always the aspect most crucial to convergence, since, as a simple example due to Wolfe [26] shows, without imposing any conditions on relaxing steps, feasible direction methods may fail to converge. The following discussion illuminates an important component in the decision to drop constraints.

Definition 3.3 (Manifold Minimization Principle)

Let x_k be an infinite sequence of **major iterates** *on some given active set A. This sequence is said to satisfy the* **manifold minimization principle** *if every limit point of the sequence is a critical point of the restricted problem $R(A)$.*

It is clear that any convergent algorithm in our framework must satisfy the manifold minimisation principle.

With this in mind we make the following definition.

Definition 3.4 (Acceptable constraint relaxation conditions)

A relaxation condition is said to be **acceptable** *if the resulting sequence of major iterates generated by our framework satisfies the manifold minimization principle.*

A number of simple, practical rules satisfy the manifold minimization condition. We give two simple examples below which are by no means exhaustive.

(i) Relax constraints only when

$$\| \text{ current reduced gradient } \| \leq \eta_k \| g(x_0) \| \quad \text{where } \eta_k \to 0 .$$

(ii) Relax constraints only after a step that satisfies both GA1 and GA2 (or their equivalent) has been taken.

We refer to rule (i) as the "**forcing sequence strategy**". It was originally proposed by Dembo [4] and has been implemented in the Primal Truncated Newton algorithm in [5] and in the PROBE algorithm described in Dembo and Tulowitzki [6].

Rule (ii) has to our knowledge, not been proposed elsewhere and is extremely easy to implement. We refer to it as the "**one sufficiently-long step on a manifold**" rule.

Both these conditions impose very weak requirements on the amount of work that needs to be done on an active set before allowing constraints to be relaxed. It is relatively straightforward to show that rule (i) satisfies the manifold minimization principle. Proposition 3.1 shows that rule (ii) does as well.

Proposition 3.1 (One sufficiently-long minor step ⇒ manifold minimization principle)

Relaxing constraints only after at least one step satisfying both GA1 and GA2 (or their equivalent) has been taken will yield an algorithm satisfying the manifold minimization principle.

Proof:

Let $\{x_k\} \to x^*$ be convergent subsequence of major iterates that lie on some (constant) active set $\mathcal{A}$. Let y_k be a minor iterate immediately preceding x_k. Then, since GA1 is satisfied by every minor iterate (i.e., the stepsize is acceptable),

$$f(x_k) \leq f(y_k) + \gamma g(y_k)^T p_k^{\mathcal{A}}$$

where $p_k^{\mathcal{A}} = (x_k - y_k)$.

Now since f is bounded below, this implies that $f(x_k) \to f(x^*)$ and hence

$$\gamma g(y_k)^\top p_k^{\mathcal{A}} \to 0 .$$

Now **assume that x^* is not a critical point of the restricted problem $R(\mathcal{A})$** which, in particular, implies that $\|g(y_k)\| \geq M > 0$ for k sufficiently large. Also

$$\lim_{k\to\infty} \|x_k - y_k\| = \lim_{k\to\infty} \|p_k^{\mathcal{A}}\| = \delta \neq 0 \text{ (by GA2, since this is a "long" step)}$$

which implies that, for k sufficiently large, using the gradient-related property of $p^{\mathcal{A}}$

$$g(y_k)^\top p_k^{\mathcal{A}} \leq -\mu \|g(y_k)\| \, \|p_k^{\mathcal{A}}\| \leq -\mu M \delta < 0$$

for some $\mu > 0$, which is a contradiction.

Q.E.D.

Remark 3.3

This lemma remains true as long as there is one sufficiently-long minor step prior to x_k. To see this, let y_k be the starting point for the long step and y_k^+ the ending point. Let all the steps from y_k^+ to x_k be "short" ones.

Then $\lim_{k\to\infty} \|x_k - y_k\| = \lim_{k\to\infty} \|y_k^+ - y_k\| \neq 0$.

Furthermore $(x_k - y_k)$ is gradient-related for sufficiently large k.

This shows that, provided $f(x_k) \leq f(y_k)$ for all k, taking a few additional short minor steps does not affect the manifold minimization properties of the "one long step on a manifold" rule.

Definition 3.5 (Global convergence)

Within the context of this paper, an algorithm is said to be globally convergent if, given an arbitrary feasible point x_0, it generates a sequence of points $\{x_k\}$ converging to a critical point of CNLP.

The manifold minimization principle is a **necessary condition for global convergence** but it is not sufficient as the following example shows.

Consider the problem:

$$\text{minimize} \quad 1/2[x_1^2 + x_2^2 + (x_3 - 1)^2]$$

$$\text{subject to} \quad x_1 \geq 0\,,\ x_2 \geq 0\,,\ x_3 \geq 0\ .$$

Now consider an algorithm that generates the following sequence of points:

x_k	$f(x_k)$	‖reduced gradient‖
(1, 0, 0)	1/2 + 1/2	1
(0, 1/2, 0)	1/8 + 1/2	1/2
(1/4, 0, 0)	1/32 + 1/2	1/4
(0, 1/8, 0)	1/128 + 1/2	1/8
.	.	.
.	.	.
.	.	.
.	.	.
(0, 0, 0)	1/2	0 (Not optimal!)

Note that it is easy to show that this algorithm drops constraints according to a forcing sequence rule (i) (page 7). Also, the subsequences restricted to the active sets A_1 (defined by $x_1 = x_3 = 0$) and A_2 (defined by $x_2 = x_3 = 0$) converge to critical points on these active sets so that the manifold minimization principle is satisfied (i.e., the constraint dropping rule is acceptable). What causes this algorithm to fail is the fact that the search directions that are generated are not gradient-related.

As we will see later in this paper, one requirement for convergence, in addition to an acceptable relaxing rule, for problems such as the above example with "box constraints", is that the search be made along projected gradient directions.

Wolfe [26] gives an example of an algorithm that uses projected-gradient directions on a box-constrained problem but fails to converge. The reason his example fails is simply that the constraint relaxation rule that is used is not acceptable since it does not satisfy the manifold minimization principle.

What then might prevent global convergence when both gradient-related directions, acceptable steps and acceptable relaxing rules are utilized? For one thing, these requirements place no restrictions other than acceptability on the relaxing steps. It is reasonable to require that these steps be "sufficiently-long", especially in the neighborhood of nonoptimal critical points. This is because acceptable relaxing rules will force all convergent subsequences to converge to critical

points for some restricted problem. We then need something that prevents us from being trapped in the neighborhood of nonoptimal critical points. This motivates the following definition.

Let $x_k^+ = x_k + p_k(\alpha_k)$ where $\|p_k\| = \alpha_k$.

Definition 3.6 (Sufficiently-long relaxing steps)

A sequence of relaxing steps is said to be **sufficiently-long** *if for every subsequence* $\{x_{k_i}\}$ *converging to a* **nonoptimal critical** *point,* $\bar{x}$*, of some restricted problem,*

$$\lim_{x_{k_i} \to \bar{x}} \|x_{k_i}^+ - x_{k_i}\| = \lim_{x_{k_i} \to \bar{x}} \alpha_{k_i} \neq 0 .$$

There are a number of assumptions and conditions that are easy to implement, each of which guarantees sufficiently-long relaxing steps. They are presented and analyzed in Section 4.

In summary, we have identified three important requirements for global convergence:

1. **gradient-related restricted and relaxing directions;**
2. **a relaxation rule that satisfies the manifold minimisation principle; and**
3. **sufficiently-long relaxing steps in the vicinity of nonoptimal critical points.**

It is fairly straightforward to see why these three conditions are sufficient for global convergence.

The manifold minimization condition ensures that all cluster points of major iterates are critical for some restricted problem $R(\mathcal{A})$. Gradient-related directions and an acceptable relaxing rule guarantee that in a neighborhood of nonoptimal critical points, for some restricted problem, constraints are dropped each time a relaxing step is taken. Sufficiently-long relaxing steps then ensure that one eventually leaves the neighborhood of any nonoptimal critical point.

This is formalized in the Theorem below.

Theorem 3.1 (Global Convergence)

Suppose f(x) is continuously differentiable and bounded below and that g(x) is Lipschitz continuous on the feasible region of CNLP. Then any limit point of an algorithm conforming to our framework, using gradient-related directions, acceptable relaxation tests and sufficiently-long relaxing steps is optimal.

Proof:

Since, by assumption, the manifold minimization condition is satisfied, it suffices to examine an infinite subsequence $\{x_k\}$ of relaxing steps.

Let x_k^+ denote the point reached after a relaxation step has been taken, that is

$$x_k^+ = x_k + p_k(\alpha_k), \quad \|p_k\| = \alpha_k .$$

Since the steps are always acceptable and in minor iterations the objective function is nonincreasing,

$$f(x_{k+1}) \leq f(x_k^+) \leq f(x_k) + \gamma g(x_k)^\top p_k(\alpha_k) .$$

Let x^* be a limit point of the algorithm. Since $f(x)$ is monotonically decreasing and bounded below, it approaches a limit point which must be $f(x^*)$. Now **assume that x^* is not an optimal (critical) point of CNLP.**

Then:

1. $\exists$ feasible, gradient-related descent steps at x^* ;
2. near x^* an acceptable relaxation test will force the algorithm to take a relaxing step;
3. long relaxing steps $\Rightarrow \lim_{k\to\infty} \alpha_k = \delta > 0$ (see Definition 3.6);
4. gradient-related steps $\Rightarrow$ for k sufficiently large, $g_k^\top p_k \leq -\mu\alpha_k$ for some $\mu > 0$;

$$\Rightarrow f(x_{k+1}) \leq f(x_k) - \gamma\delta\mu$$

$$\Rightarrow \lim_{k\to\infty} f(x_k) = -\infty$$

which contradicts the fact that f is bounded below.

Q.E.D.

Remark 3.4

The proof does not require the Lipschitz assumption. However, it might be difficult to show that steps generated by a particular algorithm are gradient-related if the gradients are not Lipschitz.

The beauty of Theorem 3.1 is that it decomposes the convergence problem into three manageable parts, namely, gradient relatedness, manifold minimization and sufficiently-long relaxing steps. Its practical value, however, hinges on the ease with which one can compute quantities which ensure that these three assumptions hold.

We have already described some simple relaxing rules that guarantee manifold minimization. Gradient-relatedness is a condition that is required in the Goldstein-Armijo framework for unconstrained optimization in order to prove that $g(x_k) \to 0$. We therefore refer readers to

discussions on this issue in Fletcher [11] and Dennis and Schnabel [9]. We have shown however that here it plays an additional role, namely, it forces constraints to be dropped in a neighborhood of a nonoptimal critical point. It only remains to be shown how the "long-relaxing step" condition may be satisfied. Doing so is instructive in that it shows the precise role played by certain assumptions in the various global convergence results in the literature.

4. Conditions that guarantee sufficiently-long relaxing steps

There are many ways to ensure sufficiently long relaxing steps. We discuss four methods that have been used widely in the literature.

1. **Nondegeneracy**, that is, at every critical point on some active set, multipliers corresponding to active inequality constraints are strictly positive (i.e., strict complementarity) ;
2. Use of a relaxing step that **drops only from among those constraints whose multiplier estimates are less than some fixed fraction of the most negative multiplier** (a particular case of this where a single constraint is dropped is described in Byrd and Schultz [3], Gill and Murray [13], Fletcher [11]; an example of the use of this rule is given in Dembo [8]) ;
3. Use of a projected gradient direction in the presence of constraints that are mutually orthogonal (see for example Bertsekas [1, 2]) ;
4. Use of ϵ-active constraint sets [2, 16, 22, 28] .

Lemma 4.1 and Proposition 4.1 below show why nondegeneracy implies sufficiently-long relaxing steps.

Lemma 4.1 (The active set settles down on nondegenerate problems)

Assume $x_k \to x^*$, a nondegenerate solution of some restricted equality-constrained problem with constraints in $A(x_k)$. Then there exists a k_0 such that for all $k \geq k_0$, $A(x_k) = A(x^*)$.

Proof:

Consider a subsequence along which $A(x_k)$ is constant. Such a subsequence exists because the number of possible active constraints is finite. Also, because the manifold minimization condition is satisfied, $x_k \to x^*$ where x^* is a critical point of the restricted problem $R(A(x_k))$.

It is clear that $A(x_k) \subset A(x^*)$. Furthermore, by the manifold minimization principle, x^* is a critical point of the problem $R(A(x_k))$. Hence the constraints in $A(x^*) \backslash A(x_k)$ are redundant and consequently have zero multipliers. By the nondegeneracy assumption this implies $A(x^*) \backslash A(x_k)$ is empty and consequently $A(x^*) = A(x_k)$.

Now assume that the active set never settles down. This implies that there are at least two active sets, $\mathcal{A}_1$ and $\mathcal{A}_2$, that recur infinitely often. But, by the above argument, $\mathcal{A}_1 = \mathcal{A}(x^*)$ and $\mathcal{A}_2 = \mathcal{A}(x^*)$ for all $k \geq k_0$. Thus $\mathcal{A}_1 = \mathcal{A}_2$ which concludes the proof.

Q.E.D.

Proposition 4.1 (Nondegeneracy ⇒ sufficiently long relaxing steps)

Assume x_k converges to some critical point, x^*, on some active set such that x^* is **not** *a first-order minimum of CNLP. Let $p_k(\alpha)$ be an unrestricted direction with $\|p_k(\alpha_k)\| = \alpha_k$. Then*

$$\lim_{\{x_k\} \to x^*} \|x_k^+ - x_k\| = \lim_{\{x_k\} \to x^*} \alpha_k \neq 0 .$$

Proof:

By examining a subsequence, if necessary, there are two possible situations:

1. GA1 and GA2 are satisfied infinitely often, or
2. GA1 is satisfied and $\alpha_k = \bar{\alpha}$ (we move to the nearest constraint along p_k) infinitely often.

If condition (1.) is true then $g(x_k)^T p_k \to 0$ by the standard argument used to show convergence for unconstrained optimization [11]. This implies that $\|g(x_k)\| \to 0$ since p_k is gradient-related. Hence x^* is a local minimum, which contradicts the assumption on x^*.

If condition (2.) is true we show by contradiction that $\alpha_k \not\to 0$.

Suppose in this case that $\alpha_k = \|x_k^+ - x_k\| \to 0$. Now since $\alpha_k = \bar{\alpha}_k$, the active set changes in going from x_k to x_k^+, that is $\mathcal{A}(x_k) \neq \mathcal{A}(x_k^+)$. In particular, $\mathcal{A}(x_k^+)$ contains at least one constraint not in $\mathcal{A}(x_k)$. But since $x_k^+ \to x^*$ **(by assumption)**, $\mathcal{A}(x_k^+) \subset \mathcal{A}(x^*) = \mathcal{A}(x_k)$ for k sufficiently large, which is a contradiction.

Q.E.D.

It is also possible to ensure sufficiently-long relaxing steps by specifying which constraints are to be dropped. This is the essential purpose of the commonly-stated rule "**drop the constraint with the most negative multiplier estimate**". We show below that this is a special case of a more general rule that permits more than one constraint to be dropped.

Suppose that in addition to the sequence of major iterates, $\{x_k\}$, the algorithm generates a corresponding sequence of multiplier estimates. Our only stipulation on these estimates is that if $\{x_k\} \to x^*$, a solution of $R(\mathcal{A}(x^*))$, then the multiplier estimates converge to the corresponding

Karush-Kuhn-Tucker multipliers for CNLP. We refer to such estimates as **consistent multiplier estimates.**

Definition 4.1 (The multiplier dropping rule)

An algorithm is said to satisfy the **multiplier dropping rule** *if it generates relaxing directions that drop only from among those constraints whose multiplier estimates are less than some fixed fraction of the most negative multiplier estimate.*

Proposition 4.2 (Multiplier dropping rule ⇒ sufficiently-long relaxing steps for functions bounded below)

Any algorithm conforming to our framework that uses consistent multiplier estimates and only relaxes constraints according to the multiplier dropping rule, will generate relaxing steps that are **sufficiently-long.**

Proof:

Assume $\{x_k\} \rightarrow x^*$ is an infinite convergent subsequence on some constant **maximal** active set $\mathcal{A}(x_k) \equiv \mathcal{A}$. By the manifold minimization condition, x^* is a solution of the restricted problems $R(\mathcal{A})$ and $R(\mathcal{A}(x^*))$. We write $\mathcal{A}_0(x^*)$ for the set of constraints in $\mathcal{A}(x^*)$ whose multipliers either have the wrong sign or are equal to zero.

Assume by way of contradiction that x^* is not optimal, which implies that $\mathcal{A}(x^*) \supset \mathcal{A}_0(x^*)$, that $\mathcal{A}(x^*) \backslash \mathcal{A}_0(x^*)$ is not empty, and that $\lim_{k \rightarrow \infty} \|x_k^+ - x_k\| = 0$, that is, relaxing steps are not "sufficiently-long". Then the GA conditions (in particular GA2) are not satisfied at x_k^+ for otherwise we could show convergence. This implies that some new constraints become active at x_k^+. Since $x_k^+ \rightarrow x^*$, we may assume that (for sufficiently large k) $\mathcal{A}(x_k^+) \subset \mathcal{A}(x^*)$, so that any new active constraints are from $\mathcal{A}(x^*) \backslash \mathcal{A}$ and, by the proof of Lemma 4.1, have zero multipliers in a neighborhood of x^*.

Since x^* is not a local minimum of CNLP, there exist some constraints at x_k^+ whose multipliers are negative and bounded away from zero. Consequently, there is a neighborhood of x^* such that a relaxing step, satisfying the multiplier dropping rule, will not drop any of the constraints in $\mathcal{A}(x^*) \backslash \mathcal{A}$. Now, if the algorithm never leaves this neighborhood, all the relaxed constraints from $\mathcal{A}$ never become active again, for then, since the additional constraints picked up from $\mathcal{A}(x^*) \backslash \mathcal{A}$ are not dropped, we will have contradicted the maximality of $\mathcal{A}$.

The above implies $\exists\ \delta > 0$ such that, for large k,

1. The algorithm leaves a δ neighborhood of x^* between the points x_k and x_{k+1}.
2. The reduced gradient is bounded away from zero at the set of points $\{x_k^+, x_k^{++}, ..., y_k\}$ where y_k is the first point outside the given δ-neighborhood (since some constraints with negative multipliers are not active).

Write x_k^0 for x_k and x_k^i for $(x_k^{i-1})^+$ and let $y_k = x_k^n$. Then

$$f(x_k) - f(x_{k+1}) \geq f(x_k) - f(y_k) \geq f(x_k^+) - f(y_k)$$
$$= \sum_{i=1}^{n-1} | f(x_k^i) - f(x_k^{i+1}) |$$
$$\geq \sum_{i=1}^{n-1} \gamma |g(x_k^i)^T p(x_k^i)| \cdot \|x_k^i - x_k^{i+1}\| \text{ by GA1.}$$

By (2.) and gradient relatedness, we may choose M such that

$$|g(x_k^i)^T p(x_k^i)| \geq M \text{ for all } i \text{ and all sufficiently large } k.$$
$$\Rightarrow f(x_k) - f(x_{k+1}) \geq \gamma M \sum_{i=1}^{n-1} \|x_k^i - x_k^{i+1}\| \geq \gamma M \|y_k - x_k^+\| .$$

Now since $x_k^+ \rightarrow x^*$ and $\|y_k - x^*\| > \delta$, the R.H.S. $\geq \gamma M \delta$ for large k, which leads to the same contradiction as in Theorem 3.1 .

Q.E.D.

5. Relationship to Other Algorithms

For our framework to be useful it should be relatively easy to show that many (or perhaps all) of the existing feasible-direction methods for CNLP may be viewed in terms of it. For our global convergence theorem to have captured the essence of the problem, it should be able to shed insight into the reason for the various conditions for global convergence that keep on recurring in the numerous convergence proofs in the literature.

To see how our analysis may be applied to existing methods we analyze two very different classes of algorithms; active set strategies and projection methods. As a prototype for active set strategies we choose the framework described in Byrd and Schultz [3]. This is because, to date, with the exception of Kovasevic [16], theirs is the least restrictive framework that we are aware of in the class of active-set methods. Our prototype for projection methods is the one described by Bertsekas [1, 2].

Byrd and Schultz [3] use gradient related, feasible restricted and relaxing steps. Manifold minimization is guaranteed by the fact that their framework only allows a **single** constraint to be dropped after an Armijo (i.e., sufficiently-long) step has been taken on some active set. Long relaxing steps are guaranteed by the multiplier dropping rule.

Using the above observations, our results show immediately how their framework may be generalized. In particular, we show that they need not restrict themselves to adding or dropping a single constraint at a time. Their proof, however, depends on this fact. Also, it becomes straightforward to obtain their theorem by mixing and matching various alternative conditions that guarantee manifold minimization and sufficiently-long relaxing steps (see Table 6.1).

Convergence Theory for Algorithms that Operate Exclusively with Relaxing Steps

There are a number of algorithms that work exclusively with relaxing steps. For example, projection methods (see Goldstein [15], Levitin and Polyak [18], Bertsekas [1, 2]) and feasible direction methods such as the Frank-Wolfe [12] algorithm, Successive Quadratic Programming [11] and Truncated Quadratic Programming [4, 7] methods. For such methods our framework has the following simple form:

Algorithmic Framework #2

START with x_0 feasible

WHILE not optimal at x_k **DO**

compute a **relaxing** step $p_k(\alpha_k)$,

and an **acceptable point** $x_k^+ = x_k + p_k(\alpha_k)$;

set $k \leftarrow k + 1$

$x_k \leftarrow x_{k-1}^+$ and **repeat.**

Theorem 5.1 (Global Convergence; Framework #2)

Suppose f(x) is continuously differentiable and bounded below on R and that g(x) is Lipschitz continuous on R^n. Then any limit point of an algorithm conforming to Framework #2 that uses gradient-related directions and steps that satisfy GA1 and GA2 (or their equivalent), converges to an optimal point.

Proof:

The proof is essentially the same as for unconstrained optimization.

Since f is bounded below, $f(x_k) \rightarrow f(x^*)$.

Assume $\{x_k\} \rightarrow x^*$, a nonoptimal point. Then by GA1

$$f(x_{k+1}) \leq f(x_k) + \gamma g(x_k)^\top p_k(\alpha_k)$$

and for k sufficiently large

$$\leq f(x_k) - \gamma \alpha_k \mu \text{ (since } p_k \text{ is gradient-related).}$$

Now since $\lim_{\{x_k\} \rightarrow x^*} \alpha_k \neq 0$ by GA2 (or its equivalent)

$$\Rightarrow \lim_{k \rightarrow \infty} f(x_k) \rightarrow -\infty \text{ , a contradiction.}$$

Q.E.D.

The first reaction one might have is, "**do such directions and steplengths exist?**". Rather than prove this directly we will exhibit some algorithms that fit in Framework #2 and satisfy the conditions of the theorem.

Consider, for example, the Frank-Wolfe algorithm [12] for linearly-constrained problems, in which a search direction is generated by solving a linear programming problem obtained by linearizing the objective function in CNLP about the point x_k. In this case p_k is given by $x^*_{LP} - x_k$ where x^*_{LP} is the optimal vertex. If a backtracking linesearch with an Armijo rule is used (as is done in Dembo and Tulowitzki [7]), then the relaxation step is long (i.e., it satisfies an equivalent GA condition). Here $\alpha_k = \|x^*_{LP} - x_k\|$ if x^*_{LP} satisfies GA1 and otherwise is the first point in a backtracking procedure (see [9] for example) that satisfies GA1. Such a point exists since p_k is a descent direction. Note that α_k is bounded away from zero at nonoptimal points x_k.

Finally, the directions that are generated are gradient-related and hence the Frank-Wolfe algorithm as implemented in [7] is globally convergent by Theorem 5.1 .

Similar arguments may be used to show that Successive and Truncated Quadratic Programming algorithms [7, 11] conform to Framework #2. For the same reasons as for Frank-Wolfe, they are convergent when a backtracking linesearch is used.

It is more interesting to see how "bending" [19] and projection algorithms [1, 2, 15, 18] may be analyzed in Framework #2. For these algorithms we compute the search direction and stepsize by iterating as follows.

Search direction and safeguarded backtracking linesearch computation for projection methods

Given d_k, a gradient-related, feasible descent direction computed at x_k , let (90° - θ) be the angle between $-g_k$ and d_k.

Define $y_k(\lambda) = [x_k + \lambda d_k]^+$ where $[x]^+$ is the projection of x onto the feasible region of CNLP (i.e., the closest feasible point to x).

Define $p_k(\lambda) = p_k(\alpha_k(\lambda)) = y_k(\lambda) - x_k$ where $\alpha_k(\lambda) = \|y_k(\lambda) - x_k\|$

and let (90° - $\theta(\lambda)$) be the angle between $p(\lambda)$ and $-g_k$.

Starting at some initial guess $\lambda_0 > 0$, backtrack (see [9] for various backtracking strategies) until:

(a) $f(y_k(\lambda)) \leq f(x_k) + \gamma g(x_k)^T p_k(\lambda)$

and

(b) $90° > \theta(\lambda) \geq \theta/N$

where $N > 1$ and $\gamma \in (0, 1)$ are fixed numbers **independent of the iteration.**

Let λ_k be the **first point in the sequence** that satisfies **both** (a) and (b). Then set

$x_k^+ = y_k(\lambda_k)$ and $p_k(\alpha_k) = x_k^+ - x_k$.

Remark 5.1

Condition (b) is a safeguard for ensuring gradient-related directions. Since d_k is a gradient-related descent direction and $p_k = \lambda d_k$ for sufficiently small λ, it is clear that there exists a λ satisfying both (a) and (b).

Remark 5.2

In order to show that projection methods are convergent using Theorem 5.1, we require p_k to be gradient-related (which it is by construction, if the above backtracking procedure is used) and α_k to be bounded away from zero at nonoptimal critical points. The modified Armijo rule proposed by Bertsekas for projected-gradient algorithms on box-constrained problems [1] fits this framework, because gradient relatedness always holds and relaxing steps are sufficiently-long for such algorithms.

Remark 5.3

The existence of such an α_k bounded away from zero follows immediately from the backtracking procedure if one assumes any of the conditions required to ensure sufficiently-long relaxing steps (see Section 4).

Remark 5.4

The drawback of projection methods is that it is very expensive to compute the projection of a point onto a general polyhedral set. Thus at first glance it might appear that they are impractical for all but problems with simple constraints such as variable bounds. However, in Dembo [8] a method is given that permits one to project cheaply onto a restriction of a general polyhedral set thereby making such methods practical for general linear constraints.

6. Summary and Conclusions

We have analyzed a number of conditions that guarantee manifold minimization and sufficiently-long relaxing steps (see Table 6.1 below). Undoubtedly there are many more. The main benefit to viewing the convergence problem as we have is that one may construct convergent algorithms by simply "mixing and matching" conditions that imply the above two properties. That is, pick any condition satisfying the manifold minimization principle and match it with any condition implying sufficiently-long relaxing steps and, provided gradient-related directions and acceptable steps are used, the resulting algorithm will be globally convergent.

In some cases, where active set determination is implicit in the search procedure, the simple but restrictive framework described in Section 5 may be more appropriate. The convergent algorithms that may be built up in this way are probably among the least restrictive (theoretically), practical convergent methods known. Although we have not addressed the problem of convergence for algorithms using trust regions, it is possible to analyze them in much the same manner as we have outlined here for linesearch-based methods.

Acknowledgements

We are grateful to Ulrich Tulowitzki for his careful reading of this paper and to Stanley Eisenstat for many stimulating discussions on this subject.

Table 6.1 Summary of Conditions Satisfying Assumptions Needed for Global Convergence

Summary of Conditions Guaranteeing:

Manifold Minimization Principle	Sufficiently-Long Relaxing Steps
1. One "long step" on a manifold (easy to implement).	1. Nondegeneracy (difficult to verify).
2. The forcing sequence strategy (easy to implement).	2. ϵ-active sets (easy to implement for some problems).
3. For $f(x)$ quadratic; exact minimization on some manifold (impractical for large problems).	3. Use of a projected gradient direction on problems with constraints that are mutually orthogonal (practical for large problems).
4. A projected gradient direction coupled with a modified-Armijo linesearch (easy to implement).	4. A projected gradient direction coupled with a modified-Armijo linesearch (easy to implement).
5. Our safeguarded backtracking procedure for projection methods (easy to implement and guarantees gradient-related steps).	5. The multiplier dropping rule (easy to implement).

References

1. Bertsekas, D. P., "On the Goldstein-Levitin-Polyak Gradient Projection Method", IEEE Transactions on Automatic Control, *21*, pp. 174-184 (1976).
2. Bertsekas, D. P., Constrained Optimization and Lagrange Multiplier Methods, Academic Press, New York (1982).
3. Byrd, R. H. and G. A. Shultz, "A Practical Class of Globally-Convergent Active-Set Strategies for Linearly-Constrained Optimization", Research Report CU-CS-238-82, Computer Science Department, University of Colorado at Boulder, September 1982.
4. Dembo, R. S., "Progress in Large-Scale Nonlinear Optimization", presented at the XI. International Symposium on Mathematical Programming, Bonn, August 1982.
5. Dembo, R. S., "NLPNET - A Code for the Solution of Nonlinear Network Optimization Problems", School of Organization and Management Working Paper Series B #70, Yale University, 1983.
6. Dembo, R. S., and U. Tulowitzki, "On the Minimization of a Quadratic Subject to Box Constraints", School of Organization and Management Working Paper Series B #71, Yale University, 1983.
7. Dembo, R. S., and U. Tulowitzki, "Computing Equilibria on Large Multicommodity Networks; an Application of Truncated Quadratic Programming", School of Organization and Management Working Paper Series B #65, Yale University, 1983.
8. Dembo, R. S., "A Primal Truncated Newton Algorithm with Application to Large-Scale Nonlinear Network Optimization", School of Organization and Management Working Paper Series B #72, Yale University, 1983.
9. Dennis, J. E. Jr., and R. B. Schnabel, Numerical Methods for Nonlinear Equations and Unconstrained Optimization, Prentice-Hall (1983).
10. Fletcher, R., "Minimizing General Functions Subject to Linear Constraints", in F. A. Lootsma, Ed., Numerical Methods for Nonlinear Optimization, Academic Press, London (1972).
11. Fletcher, R., Practical Methods of Optimization, Vol. 2, John Wiley and Sons, New York, pp. 113-117 (1981).
12. Frank, M., and P. Wolfe, "An Algorithm for Quadratic Programming", Naval Research Logistics Quarterly, *3*, pp. 95-110 (1956).
13. Gill, P. E., and W. Murray, Numerical Methods in Constrained Optimization, Academic Press, New York (1974).
14. Goldfarb, D., "Extension of Davidon's Variable Metric Method under Linear Inequality and Equality Constraints", SIAM Journal of Applied Mathematics, *17*, pp. 739-764 (1972).
15. Goldstein, A. A., "Convex Programming in Hilbert Space", Bulletin of the American Mathematical Society, *70*, No. 5, pp. 709-710 (1964).
16. Kovacevic, V., "A Convergence Theory for Linearly-Constrained Nonlinear Programming Problems", PhD. Thesis, University of Stuttgart (1977).
17. Lasdon, L.S., A.D. Waren, A. Jain and M.W. Ratner, "Design and Testing of a

Generalized Reduced Gradient Code for Nonlinear Programming", ACM Trans. Math. Software, *4*, pp. 34-50 (1978).

18. Levitin, E. S., and B. T. Polyak, "Constrained Minimization Problems", USSR Comput. Math. Phys., *6*, pp. 1-50 (1966).
19. McCormick, G. P., "Anti-Zigzagging by Bending", Management Science, *15*, pp. 315-320 (1969).
20. McCormick, G. P., "The Variable Reduction Method for Nonlinear Programming", Management Science, *17*, pp. 146-160 (1970).
21. Ortega, J. M., and W. C. Rheinboldt, Iterative Solution of Nonliner Equations in Several Variables, Academic Press, New York (1970).
22. Polak, E., Computational Methods in Optimization: A Unified Approach, Academic Press, New York (1971).
23. Rauch, S. W., "A Convergence Theory for a Class of Nonlinear Programming Problems", SIAM Journal of Numerical Analysis, *1* (1973).
24. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming: Part I, Linear Constraints", SIAM Journal, *8*, pp 181-217 (1960).
25. Topkis, D. M. and A. F. Veinott, "On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming", SIAM Journal on Control, *2*, (1967).
26. Wolfe, P., "On the Convergence of Gradient Methods Under Constraint", IBM Journal of Research and Development, *16*, pp. 407-411 (1972).
27. Zangwill, W. I., Nonlinear Programming, Prentice Hall (1969).
28. Zoutendijk, G., "Nonlinear Programming: A Numerical Survey", SIAM Journal on Control, *1*, (1966).