Chapter 7 The Concept of an Algorithm

In the remainder of the text, we describe many algorithms for solving different classes of nonlinear programming problems. This chapter introduces the concept of an algorithm. Algorithms are viewed as point-to-set maps, and the main convergence theorem is proved utilizing the concept of a closed mapping. This theorem will be utilized in the remaining chapters to analyze the convergence of several computational schemes.

The following is an outline of the chapter:

Section 7.1: Algorithms and Algorithmic Maps This section presents algorithms as point-to-set maps and introduces the concept of a solution set.

Section 7.2: Closed Maps and Convergence We first introduce the concept of a closed map and then prove the main convergence theorem.

Section 7.3: Composition of Mappings We establish closedness of composite maps by examining closedness of the individual maps. We then discuss mixed algorithms and give a condition for their convergence.

Section 7.4: Comparison Among Algorithms Some practical factors for assessing the efficiency of different algorithms are discussed.

7.1 Algorithms and Algorithmic Maps

Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in S$, where f is the objective function and S is the feasible region. A *solution procedure*, or an *algorithm*, for solving this problem can be viewed as an iterative process that generates a sequence of points according to a prescribed set of instructions, together with a termination criterion.

The Algorithmic Map

Given a vector $\mathbf{x}_k$ and applying the instructions of the algorithm, we obtain a new point $\mathbf{x}_{k+1}$. This process can be described by an *algorithmic map* **A**. This map is generally a point-to-set map and assigns to each point in the domain X a subset of X. Thus, given the initial point $\mathbf{x}_1$, the algorithmic map generates the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots,$ where $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$ for each k. The transformation of $\mathbf{x}_k$ into $\mathbf{x}_{k+1}$ through the map constitutes an *iteration* of the algorithm.

7.1.1. Example

Consider the following problem:

$$\begin{array}{ll} \text{Minimize} & x^2 \\ \text{subject to} & x \geq 1 \end{array}$$

whose optimal solution is $\bar{x} = 1$. Let the point-to-point algorithmic map be given by $\mathbf{A}(x) = \frac{1}{2}(x + 1)$. It can be easily verified that the sequence obtained by applying the map **A**, with any starting point, converges to the optimal solution $\bar{x} = 1$. With $x_1 = 4$, the algorithm generates the sequence $\{4, 2.5, 1.75, 1.375, 1.1875, \ldots\}$, as illustrated in Figure 7.1*a*.

As another example, consider the point-to-set mapping **A**, defined by

$$\mathbf{A}(x) = \begin{cases} [1, \frac{1}{2}(x + 1)] & \text{if } x \geq 1 \\ \frac{1}{2}(x + 1), 1] & \text{if } x < 1 \end{cases}$$

As shown in Figure 7.1*b*, the image of any point x is a closed interval, and any point in that interval could be chosen as the successor of x. Starting with any point x_1, the algorithm converges to $\bar{x} = 1$. With $x_1 = 4$, the sequence $\{4, 2, 1.2, 1.1., 1.02, \ldots\}$ is a possible result of the algorithm. Unlike the previous example, other sequences could result from this algorithmic map.

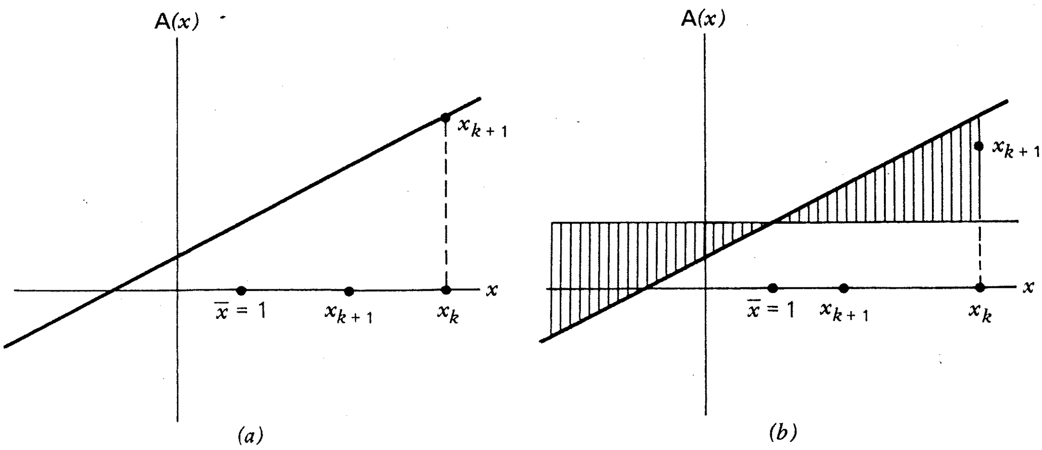

Figure 7.1 Examples of algorithmic maps.

The Solution Set and Convergence of Algorithms

Consider the following nonlinear programming problem

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in S \end{array}$$

A desirable property of an algorithm for solving the above problem is that it generates a sequence of points converging to a global optimal solution. In many cases, however, we may have to be satisfied with less favorable outcomes. In fact, as a result of nonconvexity, problem size, and other difficulties, we may stop the iterative procedure if a point belonging to a prescribed set, which we call the *solution set* Ω, is reached. The following are some typical solution sets for the above mentioned problem:

1. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}}$ is a local optimal solution of the problem$\}$.
2. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \in S, f(\bar{\mathbf{x}}) \leq b\}$, where b is an acceptable objective value.
3. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \in S, f(\bar{\mathbf{x}}) < LB + \varepsilon\}$, where $\varepsilon > 0$ is a specified tolerance and LB is a lower bound on the optimal objective value. A typical lower bound is the objective value of the Lagrangian dual problem.
4. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} \in S, f(\bar{\mathbf{x}}) - \nu^* < \varepsilon\}$, where ν^* is the known global minimum value and $\varepsilon > 0$ is specified.
5. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}}$ satisfies the KKT optimality conditions$\}$.
6. $\Omega = \{\bar{\mathbf{x}} : \bar{\mathbf{x}}$ satisfies the Fritz John optimality conditions$\}$.

Thus, in general, convergence of algorithms is made in reference to the solution set rather than to the collection of global optimal solutions. In particular, the algorithmic map $\mathbf{A}: X \to X$ is said to *converge* over $Y \subseteq X$ if, starting with any initial point $\mathbf{x}_1 \in Y$, the limit of any convergent subsequence of the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$, generated by the algorithm belongs to the solution set Ω. Letting Ω be the set of global optimal solutions in Example 7.1.1, it is obvious that the two stated algorithms are convergent over the real line.

7.2 Closed Maps and Convergence

In this section, we introduce the notion of closed maps and then prove a convergence theorem. The significance of the concept of closedness will be clear from the following example and the subsequent discussion.

7.2.1 Example

Consider the following problem:

$$\begin{array}{ll} \text{Minimize} & x^2 \\ \text{subject to} & x \geq 1 \end{array}$$

Let Ω be the set of global optimal solutions, that is, $\Omega = \{1\}$. Consider the algorithmic map defined by

$$\mathbf{A}(x) = \begin{cases} \left[\frac{3}{2} + \frac{1}{4}x, 1 + \frac{1}{2}x\right] & \text{if } x \geq 2 \\ \frac{1}{2}(x + 1) & \text{if } x < 2 \end{cases}$$

The map $\mathbf{A}$ is illustrated in Figure 7.2. Obviously, for any initial point $x_1 \geq 2$, any

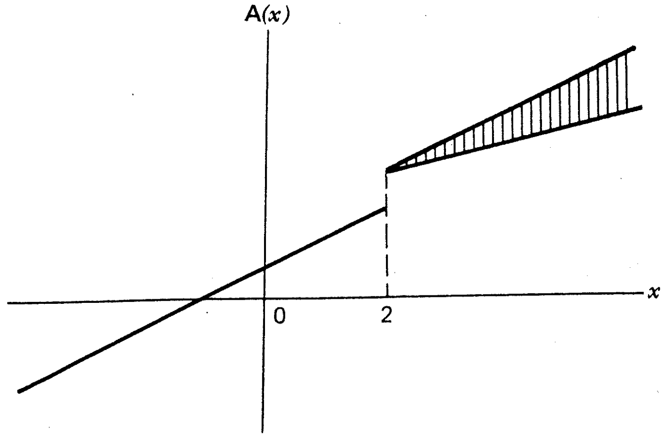

Figure 7.2 Example of a nonconvergent algorithmic map.

sequence generated by the map **A** converges to the point $\hat{x} = 2$. Note that $\hat{x} \notin \Omega$. On the other hand, for $x_1 < 2$, any sequence generated by the algorithm converges to $\bar{x} = 1$. In this example, the algorithm converges over the interval $(-\infty, 2)$ but does not converge to a point in the set Ω over the interval $[2, \infty)$.

The above example shows the significance of the initial point x_1, where convergence to a point in Ω is achieved if $x_1 < 2$ but not realized otherwise. Note that each of the algorithms in Examples 7.1.1 and 7.2.1 satisfy the following conditions:

1. Given a feasible point $x_k \geq 1$, any successor point x_{k+1} is also feasible, that is, $x_{k+1} \geq 1$.
2. Given a feasible point x_k not in the solution set Ω, any successor point x_{k+1} satisfies $f(x_{k+1}) < f(x_k)$, where $f(x) = x^2$. In other words, the objective function strictly decreases.
3. Given a feasible point x_k in the solution set Ω, that is, $x_k = 1$, the successor point is also in Ω, that is, $x_{k+1} = 1$.

Despite the above-mentioned similarities among the algorithms, the two algorithms of Example 7.1.1 converge to $\bar{x} = 1$, whereas that of Example 7.2.1 does not converge to $\bar{x} = 1$ for any initial point $x_1 \geq 2$. The reason for this is that the algorithmic map of Example 7.2.1 is not closed at $x = 2$. The notion of a closed mapping, which generalizes the notion of a continuous function, is defined below.

Closed Maps

7.2.2 Definition

Let X and Y be nonempty closed sets in E_p and E_q, respectively. Let $\mathbf{A}: X \to Y$ be a point-to-set map. The map **A** is said to be *closed* at $\mathbf{x} \in X$ if

$$\mathbf{x}_k \in X \qquad \mathbf{x}_k \to \mathbf{x}$$

$$\mathbf{y}_k \in \mathbf{A}(\mathbf{x}_k) \qquad \mathbf{y}_k \to \mathbf{y}$$

imply that $\mathbf{y} \in \mathbf{A}(\mathbf{x})$. The map **A** is said to be closed on $Z \subseteq X$ if it is closed at each point in Z.

Figure 7.2 shows an example of a point-to-set map that is not closed at $x = 2$. In particular, the sequence $\{x_k\}$ with $x_k = 2 - \frac{1}{k}$ converges to $x = 2$, and the sequence

$\{y_k\}$ with $y_k = \mathbf{A}(x_k) = \frac{3}{2} - \frac{1}{2k}$ converges to $y = \frac{3}{2}$, but $y \notin \mathbf{A}(x) = \{2\}$. Figure 7.1 shows two examples of algorithmic maps that are closed everywhere.

The Convergence Theorem

Conditions that ensure convergence of algorithmic maps are stated in Theorem 7.2.3 below. The theorem will be used in the remainder of the text to show convergence of many algorithms.

7.2.3 Theorem

Let X be a nonempty closed set in E_n, and let the nonempty set $\Omega \subseteq X$ be the solution set. Let $\mathbf{A}:X \to X$ be a point-to-set map. Given $\mathbf{x}_1 \in X$, the sequence $\{\mathbf{x}_k\}$ is generated iteratively as follows:

If $\mathbf{x}_k \in \Omega$ then stop; otherwise, let $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$, replace k by $k + 1$, and repeat

Suppose that the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ produced by the algorithm is contained in a compact subset of X, and suppose that there exists a continuous function α, called the *descent function*, such that $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$ if $\mathbf{x} \notin \Omega$ and $\mathbf{y} \in \mathbf{A}(\mathbf{x})$. If the map $\mathbf{A}$ is closed over the complement of Ω, then either the algorithm stops in a finite number of steps with a point in Ω or it generates the infinite sequence $\{\mathbf{x}_k\}$ such that

1. Every convergent subsequence of $\{\mathbf{x}_k\}$ has a limit in Ω, that is, all accumulation points of $\{\mathbf{x}_k\}$ belong to Ω.
2. $\alpha(\mathbf{x}_k) \to \alpha(\mathbf{x})$ for some $\mathbf{x} \in \Omega$.

Proof

If at any iteration a point $\mathbf{x}_k$ in Ω is generated, then the algorithm stops. Now suppose that an infinite sequence $\{\mathbf{x}_k\}$ is generated. Let $\{\mathbf{x}_k\}_{\mathcal{K}}$ be any convergent subsequence with limit $\mathbf{x} \in X$. Since α is continuous, then, for $k \in \mathcal{K}$, $\alpha(\mathbf{x}_k) \to \alpha(\mathbf{x})$. Thus, for a given $\varepsilon > 0$, there is a $k \in \mathcal{K}$ such that

$$\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}) < \varepsilon \qquad \text{for } k \geq K \text{ with } k \in \mathcal{K}$$

In particular for $k = K$, we get

$$\alpha(\mathbf{x}_K) - \alpha(\mathbf{x}) < \varepsilon \tag{7.1}$$

Now let $k > K$. Since α is a descent function, $\alpha(\mathbf{x}_k) < \alpha(\mathbf{x}_K)$, and, from (7.1), we get

$$\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}) = \alpha(\mathbf{x}_k) - \alpha(\mathbf{x}_K) + \alpha(\mathbf{x}_K) - \alpha(\mathbf{x}) < 0 + \varepsilon = \varepsilon$$

Since this is true for every $k > K$, and since $\varepsilon > 0$ was arbitrary, then

$$\lim_{k\to\infty} \alpha(\mathbf{x}_k) = \alpha(\mathbf{x}) \tag{7.2}$$

We now show that $\mathbf{x} \in \Omega$. By contradiction, suppose that $\mathbf{x} \notin \Omega$, and consider the sequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}}$. This sequence is contained in a compact subset of X and, hence, has a convergent subsequence $\{\mathbf{x}_{k+1}\}_{\bar{\mathcal{K}}}$ with limit $\bar{\mathbf{x}}$ in X. Noting (7.2), it is clear that $\alpha(\bar{\mathbf{x}}) = \alpha(\mathbf{x})$. Since $\mathbf{A}$ is closed at $\mathbf{x}$, and for $k \in \bar{\mathcal{K}}$, $\mathbf{x}_k \to \mathbf{x}$, $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$, and $\mathbf{x}_{k+1} \to \bar{\mathbf{x}}$, then $\bar{\mathbf{x}} \in \mathbf{A}(\mathbf{x})$. Therefore, $\alpha(\bar{\mathbf{x}}) < \alpha(\mathbf{x})$, contradicting the fact that $\alpha(\bar{\mathbf{x}}) =$

$\alpha(\mathbf{x})$. Thus, $\mathbf{x} \in \Omega$ and part 1 of the theorem holds true. This, coupled with (7.2), shows that part 2 of the theorem holds true, and the proof is complete.

Corollary

Under the assumptions of the theorem, if Ω is the singleton $\{\bar{\mathbf{x}}\}$, then the whole sequence $\{\mathbf{x}_k\}$ converges to $\bar{\mathbf{x}}$.

Proof

Suppose, by contradiction, that there exists an $\varepsilon > 0$ and a sequence $\{\mathbf{x}_k\}_{\mathcal{K}}$ such that

$$\|\mathbf{x}_k - \bar{\mathbf{x}}\| > \varepsilon \qquad \text{for } k \in \mathcal{K} \tag{7.3}$$

Note that there exists $\mathcal{K}' \subset \mathcal{K}$ such that $\{\mathbf{x}_k\}_{\mathcal{K}'}$ has a limit $\mathbf{x}'$. By part 1 of the theorem, $\mathbf{x}' \in \Omega$. But $\Omega = \{\bar{\mathbf{x}}\}$, and thus $\mathbf{x}' = \bar{\mathbf{x}}$. Therefore, $\mathbf{x}_k \to \bar{\mathbf{x}}$ for $k \in \mathcal{K}'$, violating (7.3). This completes the proof.

Note that if the point at hand $\mathbf{x}_k$ does not belong to the solution set Ω, then the algorithm generates a new point $\mathbf{x}_{k+1}$ such that $\alpha(\mathbf{x}_{k+1}) < \alpha(\mathbf{x}_k)$. As mentioned before, the function α is called a *descent function*. In many cases, α is chosen as the objective function f itself, and thus the algorithm generates a sequence of points with improving objective function values. Other alternative choices of the function α are possible. For instance, if f is differentiable, α could be chosen as $\alpha(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$ for an unconstrained optimization problem.

Terminating the Algorithm

As indicated in Theorem 7.2.3, the algorithm is terminated if we reach a point in the solution set Ω. In most cases, however, convergence to a point in Ω occurs only in a limiting sense, and we must resort to some practical rules for terminating the iterative procedure. The following rules are frequently used to stop a given algorithm. Here, $\varepsilon > 0$ and the positive integer N are prespecified.

1. $\|\mathbf{x}_{k+N} - \mathbf{x}_k\| < \varepsilon$
   Here, the algorithm is stopped if the distance moved after N applications of the map $\mathbf{A}$ is less than ε.

2. $\dfrac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{\|\mathbf{x}_k\|} < \varepsilon$
   Under this criterion, the algorithm is terminated if the relative distance moved during a given iteration is less than ε.

3. $\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}_{k+N}) < \varepsilon$
   Here, the algorithm is stopped if the total improvement in the descent function value after N applications of the map $\mathbf{A}$ is less than ε.

4. $\dfrac{\alpha(\mathbf{x}_k) - \alpha(\mathbf{x}_{k+1})}{|\alpha(\mathbf{x}_k)|} < \varepsilon$
   If the relative improvement in the descent function value during any given iteration is less than ε, then the termination criterion is realized.

5. $\alpha(\mathbf{x}_k) - \alpha(\bar{\mathbf{x}}) < \varepsilon$, where $\bar{\mathbf{x}}$ belongs to Ω
This criterion for termination is suitable if $\alpha(\bar{\mathbf{x}})$ is known beforehand; for example, in unconstrained optimization, if $\alpha(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$ and $\Omega = \{\bar{\mathbf{x}}: \nabla f(\bar{\mathbf{x}}) = \mathbf{0}\}$, then $\alpha(\bar{\mathbf{x}}) = 0$.

7.3 Composition of Mappings

In most nonlinear programming solution procedures, the algorithmic maps are often composed of several maps. For example, some algorithms first find a direction $\mathbf{d}_k$ to move along and then determine the step size λ_k by solving the one-dimensional problem of minimizing $\alpha(\mathbf{x}_k + \lambda \mathbf{d}_k)$. In this case, the map **A** is composed of **MD**, where **D** finds the direction $\mathbf{d}_k$, and then **M** finds an optimal step size λ_k. It is often easier to prove that the overall map is closed by examining its individual components. In this section, the notion of composite maps is stated precisely, and then a result relating closedness of the overall map to that of its individual components is given. Finally, we discuss mixed algorithms and state conditions under which they converge.

7.3.1 Definition

Let X, Y, and Z be nonempty closed sets in E_n, E_p, and E_q, respectively. Let $\mathbf{B}: X \to Y$ and $\mathbf{C}: Y \to Z$ be point-to-set maps. The *composite map* $\mathbf{A} = \mathbf{CB}$ is defined as the point-to-set map $\mathbf{A}: X \to Z$ with

$$\mathbf{A}(\mathbf{x}) = \cup \{\mathbf{C}(\mathbf{y}): \mathbf{y} \in \mathbf{B}(\mathbf{x})\}$$

Figure 7.3 illustrates the notion of a composite map, and Theorem 7.3.2 and its corollaries give several sufficient conditions for a composite map to be closed.

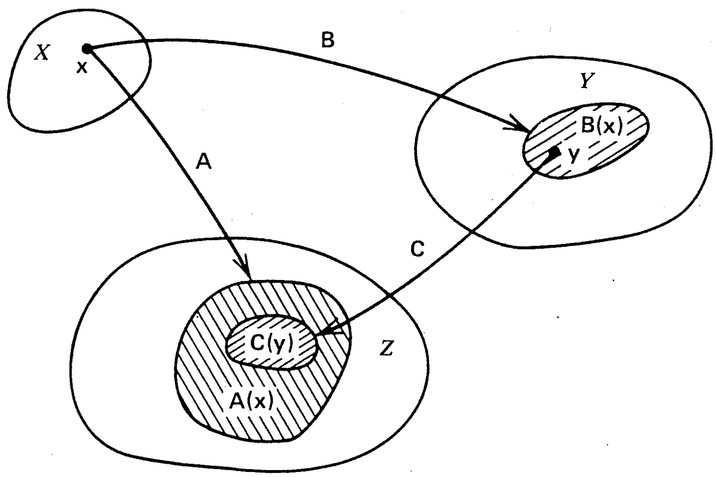

Figure 7.3 Composite maps.

7.3.2 Theorem

Let X, Y, and Z be nonempty closed sets in E_n, E_p, and E_q, respectively. Let $\mathbf{B}: X \to Y$ and $\mathbf{C}: Y \to Z$ be point-to-set maps, and consider the composite map $\mathbf{A} = \mathbf{CB}$. Suppose that **B** is closed at **x** and that **C** is closed on **B**(**x**). Furthermore, suppose that if $\mathbf{x}_k \to \mathbf{x}$ and $\mathbf{y}_k \in \mathbf{B}(\mathbf{x}_k)$, then there is a convergent subsequence of $\{\mathbf{y}_k\}$. Then, **A** is closed at **x**.

Proof

Let $\mathbf{x}_k \to \mathbf{x}$, $\mathbf{z}_k \in \mathbf{A}(\mathbf{x}_k)$, and $\mathbf{z}_k \to \mathbf{z}$. We need to show that $\mathbf{z} \in \mathbf{A}(\mathbf{x})$. By definition of **A**, for each k, there is a $\mathbf{y}_k \in \mathbf{B}(\mathbf{x}_k)$ such that $\mathbf{z}_k \in \mathbf{C}(\mathbf{y}_k)$. By assumption, there is a convergent subsequent $\{\mathbf{y}_k\}_{\mathcal{K}}$ with limit **y**. Since **B** is closed at **x**, then $\mathbf{y} \in \mathbf{B}(\mathbf{x})$. Furthermore, since **C** is closed on **B**(**x**) it is closed at **y**, and hence $\mathbf{z} \in \mathbf{C}(\mathbf{y})$. Thus, $\mathbf{z} \in \mathbf{C}(\mathbf{y}) \in \mathbf{CB}(\mathbf{x}) = \mathbf{A}(\mathbf{x})$, and hence, **A** is closed at **x**.

Corollary 1

Let X, Y, and Z be nonempty closed sets in E_n, E_p, and E_q, respectively. Let $\mathbf{B}:X \to Y$ and $\mathbf{C}:Y \to Z$ be point-to-set maps. Suppose that **B** is closed at **x**, **C** is closed on **B**(**x**), and Y is compact. Then, $\mathbf{A} = \mathbf{CB}$ is closed at **x**.

Corollary 2

Let X, Y, and Z be nonempty closed sets in E_n, E_p, and E_q, respectively. Let $\mathbf{B}:X \to Y$ be a function, and let $\mathbf{C}:Y \to Z$ be a point-to-set map. If **B** is continuous at **x**, and **C** is closed on **B**(**x**), then $\mathbf{A} = \mathbf{CB}$ is closed at **x**.

Note the importance of the assumption that a convergent subsequence $\{\mathbf{y}_k\}_{\mathcal{K}}$ exists in Theorem 7.3.2. Without this assumption, even if the maps **B** and **C** are closed, the composite map $\mathbf{A} = \mathbf{CB}$ is not necessarily closed, as shown by Example 7.3.3 below. (This example is due to Professor Jamie J. Goode.)

7.3.3 Example

Consider the maps $\mathbf{B}, \mathbf{C}:E_1 \to E_1$ defined below:

$$\mathbf{B}(x) = \begin{cases} \frac{1}{x} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

$$\mathbf{C}(y) = \{z:|z| \leq |y|\}.$$

Note that both **B** and **C** are closed everywhere. Now consider the composite map $\mathbf{A} = \mathbf{CB}$. Then, **A** is given by $\mathbf{A}(x) = \mathbf{CB}(x) = \{z:|z| \leq |\mathbf{B}(x)|\}$. From the definition of **B**, it follows that

$$\mathbf{A}(x) = \begin{cases} \{z:|z| \leq |\frac{1}{x}|\} & \text{if } x \neq 0 \\ \{0\} & \text{if } x = 0 \end{cases}$$

Note that **A** is not closed at $x = 0$. In particular, consider the sequence $\{x_k\}$, where $x_k = \frac{1}{k}$. Note that $\mathbf{A}(x_k) = \{z:|z| \leq k\}$, and hence $z_k = 1$ belongs to $\mathbf{A}(x_k)$ for each k. On the other hand, the limit point $z = 1$ does not belong to $\mathbf{A}(x) = \{0\}$. Thus, the map **A** is not closed, even though both **B** and **C** are closed. Here, Theorem 7.3.2 does not apply, since the sequence $y_k \in \mathbf{B}(x_k)$ for $x_k = \frac{1}{k}$ does not have a convergent subsequence.

Convergence of Algorithms with Composite Maps

At each iteration, many nonlinear programming algorithms use two maps **B** and **C**, say. One of the maps, **B**, is usually closed and satisfies the convergence requirements of Theorem 7.2.3. The second map, **C**, may involve any process as long as the value of the descent function does not increase. As illustrated in Exercise 7.17, the overall map may not be closed, so that Theorem 7.2.3 cannot be applied. However, as shown below, such maps do converge. Hence, such a result can be used to establish the convergence of a complex algorithm in which a step of a known convergent algorithm is interspersed at finite iteration intervals, but infinitely often over the entire algorithmic sequence. Then, by viewing the algorithm as an application of the composite map **CB**, where **B** corresponds to the step of the known convergent algorithm that satisfies the assumptions of Theorem 7.2.3 and **C** corresponds to the set of intermediate steps of the complex algorithm, the overall convergence of such a scheme would follow by Theorem 7.3.4 below. In such a context, the step of applying **B** as above is called a *spacer step*.

7.3.4 Theorem

Let X be a nonempty closed set in E_n, and let $\Omega \subseteq X$ be a nonempty solution set. Let $\alpha: E_n \to E_1$ be a continuous function, and consider the point-to-set map $\mathbf{C}: X \to X$ satisfying the following property: Given $\mathbf{x} \in X$, then $\alpha(\mathbf{y}) \leq \alpha(\mathbf{x})$ for $\mathbf{y} \in \mathbf{C}(\mathbf{x})$. Let $\mathbf{B}: X \to X$ be a point-to-set map that is closed over the complement of Ω and that satisfies $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$ for each $\mathbf{y} \in \mathbf{B}(\mathbf{x})$, if $\mathbf{x} \notin \Omega$. Now consider the algorithm defined by the composite map $\mathbf{A} = \mathbf{CB}$. Given $\mathbf{x}_1 \in X$, suppose that the sequence $\{\mathbf{x}_k\}$ is generated as follows:

If $\mathbf{x}_k \in \Omega$, then stop; otherwise, let $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$ replace k by $k + 1$, and repeat

Suppose that the set $\Lambda = \{\mathbf{x}: \alpha(\mathbf{x}) \leq \alpha(\mathbf{x}_1)\}$ is compact. Then, either the algorithm stops in a finite number of steps with a point in Ω or all accumulation points of $\{\mathbf{x}_k\}$ belong to Ω.

Proof

If at any iteration $\mathbf{x}_k \in \Omega$, then the algorithm stops finitely. Now suppose that the sequence $\{\mathbf{x}_k\}$ is generated by the algorithm, and let $\{\mathbf{x}_k\}_{\mathcal{K}}$ be a convergent subsequence, with limit $\mathbf{x}$. Thus, $\alpha(\mathbf{x}_k) \to \alpha(\mathbf{x})$ for $k \in \mathcal{K}$. Using monotonicity of α as in Theorem 7.2.3, it follows that

$$\lim_{k\to\infty} \alpha(\mathbf{x}_k) = \alpha(\mathbf{x}) \tag{7.4}$$

We want to show that $\mathbf{x} \in \Omega$. By contradiction, suppose that $\mathbf{x} \notin \Omega$, and consider the sequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}}$. By definition of the composite map $\mathbf{A}$, note that $\mathbf{x}_{k+1} \in \mathbf{C}(\mathbf{y}_k)$, where $\mathbf{y}_k \in \mathbf{B}(\mathbf{x}_k)$. Note that $\mathbf{y}_k$, $\mathbf{x}_{k+1} \in \Lambda$. Since Λ is compact, there exists an index set $\mathcal{K}' \subseteq \mathcal{K}$ such that $\mathbf{y}_k \to \mathbf{y}$ and $\mathbf{x}_{k+1} \to \mathbf{x}'$ for $k \in \mathcal{K}'$. Since $\mathbf{B}$ is closed at $\mathbf{x} \notin \Omega$, then $\mathbf{y} \in \mathbf{B}(\mathbf{x})$, and $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$. Since $\mathbf{x}_{k+1} \in \mathbf{C}(\mathbf{y}_k)$, then, by assumption, $\alpha(\mathbf{x}_{k+1}) \leq \alpha(\mathbf{y}_k)$ for $k \in \mathcal{K}'$; and, hence, by taking the limit, $\alpha(\mathbf{x}') \leq \alpha(\mathbf{y})$. Since $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$, then $\alpha(\mathbf{x}') < \alpha(\mathbf{x})$. Since $\alpha(\mathbf{x}_{k+1}) \to \alpha(\mathbf{x}')$ for $k \in \mathcal{K}'$, then $\alpha(\mathbf{x}') < \alpha(\mathbf{x})$ contradicts (7.4). Therefore, $\mathbf{x} \in \Omega$, and the proof is complete.

Minimizing Along Independent Directions

We now present a theorem that establishes convergence of a class of algorithms for solving a problem of the form: minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in E_n$. Under mild assumptions, we show that an algorithm that generates n linearly independent search directions, and obtains a new point by sequentially minimizing f along these directions, converges to a stationary point. The theorem also establishes convergence of algorithms using linearly independent and orthogonal search directions.

7.3.5 Theorem

Let $f: E_n \to E_1$ be differentiable, and consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in E_n$. Consider an algorithm whose map $\mathbf{A}$ is defined as follows. The vector $\mathbf{y} \in \mathbf{A}(\mathbf{x})$ means that $\mathbf{y}$ is obtained by minimizing f sequentially along the directions $\mathbf{d}_1, \ldots, \mathbf{d}_n$ starting from $\mathbf{x}$. Here, the search directions $\mathbf{d}_1, \ldots, \mathbf{d}_n$ may depend upon $\mathbf{x}$, and each has norm 1. Suppose that the following properties are true:

1. There exists an $\varepsilon > 0$ such that det $[\mathbf{D}(\mathbf{x})] \geq \varepsilon$ for each $\mathbf{x} \in E_n$. Here, $\mathbf{D}(\mathbf{x})$ is the $n \times n$ matrix whose columns are the search directions generated by the algorithm, and det $[\mathbf{D}(\mathbf{x})]$ denotes the determinant of $\mathbf{D}(\mathbf{x})$.
2. The minimum of f along any line in E_n is unique.

Given a starting point $\mathbf{x}_1$, suppose that the algorithm generates the sequence $\{\mathbf{x}_k\}$ as follows. If $\nabla f(\mathbf{x}_k) = \mathbf{0}$, then the algorithm stops with $\mathbf{x}_k$; otherwise, $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$, k is replaced by $k + 1$, and the process is repeated. If the sequence $\{\mathbf{x}_k\}$ is contained in a compact subset of E_n, then each accumulation point $\mathbf{x}$ of the sequence $[\mathbf{x}_k\}$ must satisfy $\nabla f(\mathbf{x}) = \mathbf{0}$.

Proof

If the sequence $\{\mathbf{x}_k\}$ is finite, then the result is immediate. Now suppose that the algorithm generates the infinite sequence $\{\mathbf{x}_k\}$.

Let $\mathcal{K}$ be an infinite sequence of positive integers, and suppose that the sequence $\{\mathbf{x}_k\}_{\mathcal{K}}$ converges to a point $\mathbf{x}$. We need to show that $\nabla f(\mathbf{x}) = \mathbf{0}$. Suppose by contradiction that $\nabla f(\mathbf{x}) \neq \mathbf{0}$, and consider the sequence $\{\mathbf{x}_{k+1}\}_{\mathcal{K}}$. By assumption, this sequence is contained in a compact subset of E_n; and, hence, there exists $\mathcal{K}' \subseteq \mathcal{K}$ such that $\{\mathbf{x}_{k+1}\}_{\mathcal{K}'}$ converges to $\mathbf{x}'$. We shall first show that $\mathbf{x}'$ can be obtained from $\mathbf{x}$ by minimizing f along a set of n linearly independent directions.

Let $\mathbf{D}_k$ be the $n \times n$ matrix whose columns $\mathbf{d}_{1k}, \ldots, \mathbf{d}_{nk}$ are the search directions generated at iteration k. Thus, $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{D}_k\boldsymbol{\lambda}_k = \mathbf{x}_k + \sum_{j=1}^{n} \mathbf{d}_{jk}\lambda_{jk}$, where λ_{jk} is the distance moved along $\mathbf{d}_{jk}$. In particular, letting $\mathbf{y}_{1k} = \mathbf{x}_k$, $\mathbf{y}_{j+1,k} = \mathbf{y}_{jk} + \lambda_{jk}\mathbf{d}_{jk}$ for $j = 1, \ldots, n$ it follows that $\mathbf{x}_{k+1} = \mathbf{y}_{n+1,k}$, and

$$f(\mathbf{y}_{j+1,k}) \leq f(\mathbf{y}_{jk} + \lambda \mathbf{d}_{jk}) \qquad \text{for all } \lambda \in E_1 \text{ and } j = 1, \ldots, n \tag{7.5}$$

Since det $[\mathbf{D}_k] \geq \varepsilon > 0$, $\mathbf{D}_k$ is invertible, so that $\boldsymbol{\lambda}_k = \mathbf{D}_k^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$. Since each column of $\mathbf{D}_k$ has norm 1, there exists $\mathcal{K}'' \subseteq \mathcal{K}'$ such that $\mathbf{D}_k \to \mathbf{D}$. Since det $[\mathbf{D}_k] \geq \varepsilon$ for each k, det $[\mathbf{D}] \geq \varepsilon$, so that $\mathbf{D}$ is invertible. Now, for $k \in \mathcal{K}''$, $\mathbf{x}_{k+1} \to \mathbf{x}'$, $\mathbf{x}_k \to \mathbf{x}$, $\mathbf{D}_k \to \mathbf{D}$, so that $\boldsymbol{\lambda}_k \to \boldsymbol{\lambda}$, where $\boldsymbol{\lambda} = \mathbf{D}^{-1}(\mathbf{x}' - \mathbf{x})$. Therefore, $\mathbf{x}' = \mathbf{x} + \mathbf{D}\boldsymbol{\lambda} = \mathbf{x} + \sum_{j=1}^{n} \mathbf{d}_j\lambda_j$. Let $\mathbf{y}_1 = \mathbf{x}$, and for $j = 1, \ldots, n$, let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j\mathbf{d}_j$, so that $\mathbf{x}' = \mathbf{y}_{n+1}$. To show that $\mathbf{x}'$ is obtained from $\mathbf{x}$ by minimizing f sequentially along $\mathbf{d}_1, \ldots, \mathbf{d}_n$, it suffices to show that

$$f(\mathbf{y}_{j+1}) \le f(\mathbf{y}_j + \lambda \mathbf{d}_j) \qquad \text{for all } \lambda \in E_1 \text{ and } j = 1, \ldots, n \tag{7.6}$$

Note that $\lambda_{jk} \to \lambda_j$, $\mathbf{d}_{jk} \to \mathbf{d}_j$, $\mathbf{x}_k \to \mathbf{x}$, and $\mathbf{x}_{k+1} \to \mathbf{x}'$ as $k \in \mathcal{K}''$ approaches ∞, so that $\mathbf{y}_{jk} \to \mathbf{y}_j$ for $j = 1, \ldots, n+1$ as $k \in \mathcal{K}''$ approaches ∞. By continuity of f, then, (7.6) follows from (7.5). We have thus shown that $\mathbf{x}'$ is obtained from $\mathbf{x}$ by minimizing f sequentially along the directions $\mathbf{d}_1, \ldots, \mathbf{d}_n$.

Obviously, $f(\mathbf{x}') \le f(\mathbf{x})$. First, consider the case $f(\mathbf{x}') < f(\mathbf{x})$. Since $\{f(\mathbf{x}_k)\}$ is a nonincreasing sequence, and since $f(\mathbf{x}_k) \to f(\mathbf{x})$ as $k \in \mathcal{K}$ approaches ∞, $\lim_{k\to\infty} f(\mathbf{x}_k) = f(\mathbf{x})$. This is impossible, however, in view of the fact that $\mathbf{x}_{k+1} \to \mathbf{x}'$ as $k \in \mathcal{K}'$ approaches ∞ and the assumption that $f(\mathbf{x}') < f(\mathbf{x})$. Now consider the case $f(\mathbf{x}') = f(\mathbf{x})$. By property 2 of the theorem, and since $\mathbf{x}'$ is obtained from $\mathbf{x}$ by minimizing f along $\mathbf{d}_1, \ldots, \mathbf{d}_n$, $\mathbf{x}' = \mathbf{x}$. This further implies that $\nabla f(\mathbf{x})^t \mathbf{d}_j = 0$ for $j = 1, \ldots, n$. Since $\mathbf{d}_1, \ldots, \mathbf{d}_n$ are linearly independent, $\nabla f(\mathbf{x}) = \mathbf{0}$, contradicting our assumption. This completes the proof.

Note that no closedness or continuity assumptions are made on the map providing the search directions. It is only required that the search directions used at each iteration be linearly independent and that as these directions converge, the limiting directions must also be linearly independent. Obviously this holds true if a fixed set of linearly independent search directions are used at every iteration. Alternatively, if the search directions used at each iteration are mutually orthogonal, and each has norm 1, then the search matrix $\mathbf{D}$ satisfies $\mathbf{D}^t\mathbf{D} = \mathbf{I}$. Therefore, det $[\mathbf{D}] = 1$, so that condition 1 of the theorem holds true.

Also note that condition 2 in the statement of the theorem is used to ensure the following property. If a differentiable function f is minmized along n independent directions starting from a point $\mathbf{x}$ and resulting in $\mathbf{x}'$, then $f(\mathbf{x}') < f(\mathbf{x})$, provided that $\nabla f(\mathbf{x}) \ne \mathbf{0}$. Without assumption 2, this is not true, as evidenced by $f(x_1, x_2) = x_2(1 - x_1)$. If $\mathbf{x} = (0, 0)^t$, then minimizing f starting from $\mathbf{x}$ along $\mathbf{d}_1 = (1, 0)^t$ and then along $\mathbf{d}_2 = (0, 1)^t$ could produce the point $\mathbf{x}' = (1, 1)^t$, where $f(\mathbf{x}') = f(\mathbf{x}) = 0$, even though $\nabla f(\mathbf{x}) = (0, 1)^t \ne (0, 0)^t$.

7.4 Comparison Among Algorithms

In the remainder of the text, we discuss several algorithms for solving different classes of nonlinear programming problems. This section discusses some important factors that must be considered when assessing the effectiveness of these algorithms and when comparing them. These factors are (1) generality, reliability, and precision; (2) sensitivity to parameters and data; (3) preparational and computational effort; and (4) convergence.

Generality, Reliability, and Precision

Different algorithms are designed for solving various classes of nonlinear programming problems, such as unconstrained optimization problems, problems with inequality constraints, problems with equality constraints, and problems with both types of constraints. Within each of these classes, different algorithms make specific assumptions about the problem structure. For example, for unconstrained optimization problems, some procedures assume that the objective function is differentiable, whereas other algorithms do not make this assumption and rely primarily on functional evaluations