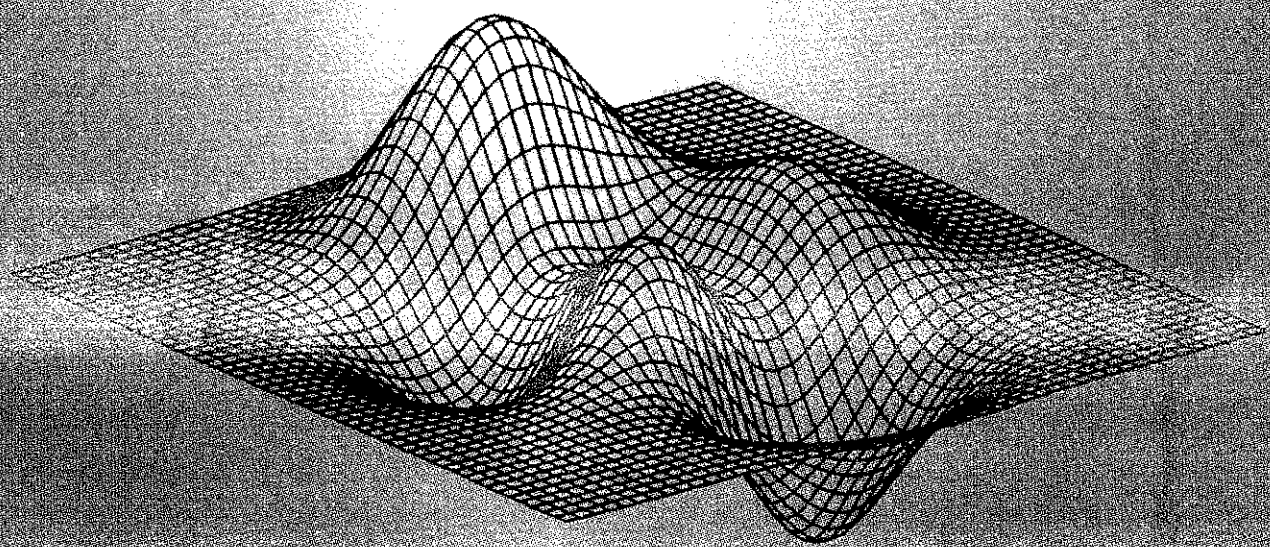
NONCONVEX OPTIMIZATION AND ITS APPLICATIONS

Introduction to Global Optimization

2nd Edition

Reiner Horst, Panos M. Pardalos and Nguyen V. Thoai



Kluwer Academic Publishers

Nonconvex Optimization and Its Applications

Volume 48

The titles published in this series are listed at the end of this volume.

Introduction to Global Optimization

2nd Edition

by

Reiner Horst
Trier University

Panos M. Pardalos
University of Florida

and

Nguyen V. Thoai
Trier University

KLUWER ACADEMIC PUBLISHERS
DORDRECHT / BOSTON / LONDON

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 0-7923-6574-7 (HB)
ISBN 0-7923-6756-1 (PB)

Published by Kluwer Academic Publishers,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

Sold and distributed in North, Central and South America
by Kluwer Academic Publishers,
101 Philip Drive, Norwell, MA 02061, U.S.A.

In all other countries, sold and distributed
by Kluwer Academic Publishers,
P.O. Box 322, 3300 AH Dordrecht, The Netherlands.

Printed on acid-free paper

Printed in the Netherlands.

"The hidden harmony is better than the obvious"

"People do not understand how that which is a variance with itself agrees with itself. There is a harmony in the bending back, as in the case of the bow and the lyre"

– Heraclitus

Contents

Preface to the Second Edition

In this edition, the scope and character of the monograph did not change with respect to the first edition. Taking into account the rapid development of the field, we have, however, considerably enlarged its contents. Chapter 4 includes two additional sections 4.4 and 4.6 on theory and algorithms of D.C. Programming. Chapter 7, on Decomposition Algorithms in Nonconvex Optimization, is completely new. Besides this, we added several exercises and corrected errors and misprints in the first edition.

We are grateful for valuable suggestions and comments that we received from several colleagues.

R. Horst, P.M. Pardalos and N.V. Thoai March 2000

Chapter 5

Lipschitz Optimization

5.1 Lipschitz Functions

In this chapter we present an introduction into the optimization techniques of Lipschitz functions with known Lipschitz constant.

Let, as usual, for $z \in \mathbb{R}^n$, $\| z \|$ denote the Euclidean norm. A real-valued function f is called *Lipschitzian* (or a *Lipschitz function*) on a set $P \subset \mathbb{R}^n$ if there is a (Lipschitz) constant $L = L(f, P) > 0$ such that

$$| f(x_2) - f(x_1) | \leq L \| x_2 - x_1 \| \text{ for all } x_1, x_2 \in P. \tag{5.1}$$

Obviously, if f is Lipschitzian on P with constant L, then f is also Lipschitzian on P with all constants $L' > L$.

In some applications one considers the relation (5.1) with respect to other ℓ_p–norms ($1 \leq p \leq \infty$) defined by

$$\| z \|_p = \left(\sum_{i=1}^{n} | z_i |^p \right)^{1/p}, \ 1 \leq p \leq \infty, \ \text{where} \ \| z \|_\infty = \max_{i=1,\ldots,n} | z_i | .$$

It follows from (5.1) that a Lipschitz function is continuous on P. When P is a real interval the Lipschitz constant L provides an upper bound for the absolute value of the slope of any line joining two points on the graph of f. Since, for continuous functions, this slope is not necessarily bounded, we see that a continuous function is not necessarily Lipschitzian (take, for example, $f(x) = \sqrt{x}$, $P = [0, 1]$).

Proposition 5.1. *Let $P \subset \mathbb{R}^n$ be convex, and let f be continuously differentiable on an open set containing P with bounded gradient on P. Then f is Lipschitzian on P with constant*

$$L = \sup\{\| \nabla f(x) \| : x \in P\}. \tag{5.2}$$

Proof: For every pair $x_1, x_2 \in P$ there exists $z = x_2 + \lambda(x_1 - x_2) \in P$, $0 < \lambda < 1$, such that $| f(x_2) - f(x_1) | = | (x_2 - x_1)^T \nabla f(z) |$ (by the mean–value theorem). Clearly, $| (x_2 - x_1)^T \nabla f(z) | \leq \sup\{\| \nabla f(z) \| : z \in P\} \| x_2 - x_1 \|$. □

In the sequel, we will only consider bounded closed sets P so that boundedness of the gradient ∇f follows from continuity of ∇f, and the supremum in (5.2) can be replaced by the maximum of $\| \nabla f(x) \|$ over P. In general, finding a good upper bound of (5.2) can be a difficult task. In most applications, however, where Lipschitzian optimization techniques are used, the sets P are successively refined rectangles or simplices for which adaptive approximations of L can be used. A quite useful technique to obtain estimates of (5.2) over rectangles is provided by interval analysis (e.g., Ratschek and Rokne (1988)). Other helpful techniques will be given below.

Example 5.1. Let $f(x) = \sum_{k=1}^{n} ((1/2)p_k x_k^2 + q_k x_k + r_k)$ and $P = \{x \in \mathbb{R}^n : a_k \leq x_k \leq b_k,\ k = 1, \ldots, n\}$ with given real numbers $p_k \neq 0$, $q_k, r_k, a_k < b_k$ $(k = 1, \ldots, n)$. Then the relation (5.2) yields

$$L = \left[\sum_{k \in I_1} (p_k a_k + q_k)^2 + \sum_{k \in I_2} (p_k b_k + q_k)^2 \right]^{1/2}$$

where

$$\begin{aligned} I_1 &= \{k : -q_k/p_k \geq (1/2)(a_k + b_k)\}, \\ I_2 &= \{k : -q_k/p_k < (1/2)(a_k + b_k)\} \end{aligned}$$

(Exercise).

Example 5.2. Let $P := [a, b]$, $a < b$, be a real interval and let $f(x)$ be a polynomial with the derivative $f'(x)$ decomposed as

$$f'(x) = r(x) - q(x) + c = r_e(x) + r_o(x) - q_e(x) - q_o(x) + c,$$

where $r(x), q(x), r_e(x), r_o(x), q_e(x)$ and $q_o(x)$ are polynomials with positive coefficients, and c is a constant. Moreover, $r_e(x)$ and $q_e(x)$ have only even

exponents, whereas $r_o(x)$ and $q_o(x)$ have only odd exponents. Then $L = \max\{L_1, L_2\}$ is a Lipschitz constant of $f(x)$ over P, where

$$L_1 = r_e(\max\{-a, b\}) + r_o(b) - q_e(\min\{-a, 0, b\}) - q_o(a) + c$$

and

$$L_2 = \mid r_e(\min\{-a, 0, b\}) + r_o(a) - q_e(\max\{-a, b\}) - q_o(b) + c \mid .$$

If $a > 0$, then (5.2) yields

$$L = \max\{r(b) - q(a) + c, \mid r(a) - q(b) + c \mid\}$$

(Exercise).

Proposition 5.2. *Let f, g, f_i $(i = 1, \ldots, m)$ be Lipschitz functions on the compact set $P \subset \mathbb{R}^n$. Then the following assertions hold:*

(i) *Every linear combination of f_i $(i = 1, \ldots, m)$ is Lipschitzian on P,*

(ii) $\max\limits_{i=1,\ldots,m} f_i$ *and* $\min\limits_{i=1,\ldots,m} f_i$ *are Lipschitzian on P,*

(iii) *the product $f \cdot g$ is Lipschitzian on P.*

Proof: The straightforward proof is constructive, i.e., it provides a Lipschitz constant for the functions in (i) – (iii) whenever the Lipschitz constants of the functions f, g, f_i are known. For example, when the functions f_i are Lipschitzian on P with Lipschitz constants L_i, then we see from

$$\mid \max_i f_i(x_2) - \max_i f_i(x_1) \mid \leq \max_i \mid f_i(x_2) - f_i(x_1) \mid$$

$$\leq \max_i\{L_i \parallel x_2 - x_1 \parallel\} = \left(\max_i L_i\right) \parallel x_2 - x_1 \parallel$$

that $f(x) = \max\limits_{i=1,\ldots,m} f_i(x)$ is Lipschitzian on P with Lipschitz constant $L = \max\limits_{i=1,\ldots,m} L_i$. The proof for the remaining cases is left to the reader as an exercise. □

Proposition 5.3. *Let f_i be Lipschitzian on $P \subset \mathbb{R}^n$ with Lipschitz constants L_i $(i = 1, \ldots, m)$. Then, for each $1 \leq p \leq \infty$, $f(x) = \left(\sum\limits_{i=1}^{m} \mid f_i(x) \mid^p\right)^{1/p}$ is Lipschitzian on P with Lipschitz constant $\sum\limits_{i=1}^{m} L_i$.*

Proof: Let $F = (f_1, \ldots, f_m)^T$. Then $f(x)$ is the ℓ_p–norm $\| F(x) \|_p$ of F in $\mathbb{R}^m$, and we obtain the following chain of relations:

$$\begin{aligned} &|\| F(x_2) \|_p - \| F(x_1) \|_p| \leq \| F(x_2) - F(x_1) \|_p \leq \| F(x_2) - F(x_1) \|_1 \\ &= \sum_{i=1}^{m} | f_i(x_2) - f_i(x_1) | \leq \sum_{i=1}^{m} L_i \| x_2 - x_1 \| = \| x_2 - x_1 \| \ (\sum_{i=1}^{m} L_i), \end{aligned}$$

where $\| \cdot \|$ denotes the Euclidean norm in $\mathbb{R}^n$. The first inequality is an immediate consequence of the triangle inequality, and hence holds for any norm. The second inequality uses the fact that

$$\| z \|_p \leq \| z \|_1 \quad \text{for any } z \in \mathbb{R}^m,\ 1 \leq p \leq \infty$$

(Exercise). □

5.2 Lipschitz Optimization Problems

Let $D \subset \mathbb{R}^n$ be a compact set, and let f be a real valued Lipschitz function on a compact set $P \supset D$. When $D := \{x \in \mathbb{R}^n : g_j(x) \leq 0\ (j = 1, \ldots, m)\}$, where the functions g_j are Lipschitzian on P, the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in D \end{aligned}$$

is called a *Lipschitz optimization problem.* In this introduction we consider the following subclasses:

a) the feasible set *is a rectangle* $\{x \in \mathbb{R}^n : a \leq x \leq b\}$, $a < b \in \mathbb{R}^n$;

b) the feasible set *is an n–simplex* $[v_0, \ldots, v_n]$ with vertices $v_0, \ldots, v_n$;

c) the feasible set *is a polytope* $\{x \in \mathbb{R}^n : Ax \leq b\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$;

d) the feasible set is of the form $\{x : g_i(x) \leq 0,\ i = 1, \ldots, m\}$, where g_i are Lipschitz functions with known Lipschitz constants.

Next we briefly indicate some large fields of applications which lead to Lipschitz optimization problems.

5.2.1 Approximation problems

The best approximation of a given function φ by means of a family K of simpler functions which depends on some parameters $x_1, \ldots, x_n$ is an essential problem in numerical analysis. A well–known example is the representation of a (complicated) function φ on a computer, where φ is replaced by an approximating polynomial or a suitable rational function the values of which are easily computable. The difference (error bound) between φ and the approximating functions is measured by a suitable norm.

Let, for example, $S \subset \mathbb{R}^m$ be a closed, bounded convex set with nonempty interior, and let $\varphi(y)$ be a given continuous function on S. We want to approximate φ on S by a function $\overline{\Psi}(y)$, which is chosen from a class $K = \{\Psi(y; x_1, \ldots, x_n)\}$ of continuous functions $\Psi(y; x_1, \ldots, x_n)$ by fixing the parameters $x_1, \ldots, x_n$, in such a way that for a given norm $\| \cdot \|$ on the space of continuous functions on S we have:

$$\| \varphi - \overline{\Psi} \| = \min_{\Psi \in K} \| \varphi - \Psi \| . \tag{5.3}$$

Let $x = (x_1, \ldots, x_n)^T$ and suppose that the parameter–vector is to be chosen from a rectangle $R \subset \mathbb{R}^n$. Moreover, assume that $\Psi : S \times R \to \mathbb{R}$ is Lipschitzian on R for each fixed $y \in S$ with Lipschitz constant $L(y)$ such that

$$L = \max_{y \in S} L(y) \tag{5.4}$$

exists.

In most applications, one chooses an L_p–norm defined by

$$L_p(\varphi - \Psi) = [\int_S | (\varphi(y) - \Psi(y; x) |^p \, dy]^{1/p} \tag{5.5}$$

for some $1 \leq p \leq \infty$.

Most well–known special cases are

$$L_\infty = \max_{y \in S} | \varphi(y) - \Psi(y; x) | \quad \text{(Chebyshev–approximation)}$$

and

$$L_2 = [\int_S (\varphi(y) - \Psi(y; x))^2 dy]^{1/2} \quad \text{(least–squares approximation).}$$

In this setting, problem (5.3) becomes

$$\min_{x\in R}[\int_S \mid \varphi(y) - \Psi(y;x) \mid^p dy]^{1/p}. \tag{5.6}$$

It is easy to see that the objective function in (5.6) is Lipschitzian on R, since for $x_1, x_2 \in R$ one has (in virtue of the well–known Minkowski–inequality (triangle inequality for the L_p norms)) the following:

$$\begin{aligned}
&\mid [\int_S \mid \varphi(y) - \Psi(y;x_2) \mid^p dy]^{1/p} - [\int_S (\mid \varphi(y) - \Psi(y;x_1)^p dy]^{1/p} \mid \\
&\leq [\int_S \mid \Psi(y;x_2) - \Psi(y;x_1) \mid^p dy]^{1/p} \leq (\text{vol } (S))^{1/p} \max_{y\in S} \mid \Psi(y;x_2) - \Psi(y;x_1) \mid \\
&\leq (\text{vol } (S))^{1/p} \max_{y\in S}(L(y) \parallel x_2 - x_1 \parallel_2) = L \ (\text{vol } (S))^{1/p} \parallel x_2 - x_1 \parallel_2,
\end{aligned}$$

where $\parallel \cdot \parallel_2$ is the Euclidean norm in $\mathbb{R}^n$ and vol (S) is the volume of S. Clearly, in the above problem, the rectangle R can be replaced by more general sets.

Note that the above approximation problems are also closely related to various statistical parameter estimation (curve fitting) problems.

Example 5.3. For $S = [a,b] \subset \mathbb{R}$, $0 < a < b$, and

$$K = \{\Psi(y;x) = x_4y^3 + x_3y^2 + x_2y + x_1;\ -c \leq x_i \leq +c \ (i = 1,\ldots,4)\},\ c > 0,$$

one has the problem of approximating a continuous function on the interval $[a,b]$ by a polynomial of degree at most three with coefficients in $[-c,+c]$. The functions $\Psi(y;x)$ are linear in x, and hence Lipschitzian with $L(y) = (y^6+y^4+y^2+1)^{1/2}$. We obtain $L = (b^6+b^4+b^2+1)^{1/2}$ and vol $(S) = b-a$.

5.2.2 Systems of nonlinear equations and inequalities

Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a vector–valued mapping and R be a rectangle in $\mathbb{R}^n$. Suppose that, for $i = 1,\ldots,m$, the components f_i of f are Lipschitzian on R with Lipschitz constants L_i. Let $\parallel \cdot \parallel_p$ denote an ℓ_p–norm in $\mathbb{R}^m$. Then $x^* \in R$ solves the *system of equations* $f_i(x) = 0$ $(i = 1,\ldots,m)$ if and only if

$$\parallel f(x^*) \parallel_p = \min_{x\in R} \parallel f(x) \parallel_p = 0, \tag{5.7}$$

because for any norm we have $\parallel z \parallel = 0$ if and only if $z = 0$.

We know from Proposition 5.3 that the objective function in (5.7) is Lipschitzian on R with Lipschitz constant $L = \sum_{i=1}^{m} L_i$.

Likewise, we see that $x^* \in R$ solves the *system of inequalities* $f_i(x) \leq 0$ $(i = 1, \ldots, m)$ if and only if

$$\max_{i=1,\ldots,m} f_i(x^*) \leq 0, \tag{5.8}$$

where $\max_{i=1,\ldots,m} f_i(x)$ is Lipschitzian on R with constant $L = \max_{i=1,\ldots,m} L_i$ (cf. Proposition 5.2). Therefore, the above system of inequalities can be investigated by applying an algorithm for the Lipschitzian optimization problem

$$\min_{x \in R} \max_{i=1,\ldots,m} f_i(x) \tag{5.9}$$

until a point $x^* \in R$ satisfying (5.8) is detected or the optimal objective function value in (5.9) is found to be positive (which means that the system has no solution in R).

Notice that a system of inequalities can be transformed into a Lipschitzian optimization problem in a number of alternative ways. For example, $x^* \in R$ solves the system $f_i(x) \leq 0$ $(i = 1, \ldots, m)$ if and only if, for arbitrary $p > 0$, x^* solves

$$\min_{x \in R} \sum_{i=1}^{m} (\max\{f_i(x), 0\})^p = 0. \tag{5.10}$$

It is clear, that in the above discussion, the rectangle R can again be replaced by different sets. In chemical factories, for example in oil refineries, one has the problem of identifying mixture products, each represented by a vector $x \in \mathbb{R}^n$, which meets certain requirements described by a set of inequalities $f_i(x) \leq 0$, $i = 1, \ldots, m$. The set of possible mixtures is mathematically defined by the simplex $S = \{x \in \mathbb{R}^n : \sum_{j=1}^{n} x_j = 1, x_j \geq 0, j = 1, \ldots, n\}$, and we are led to problems of the type (5.9), (5.10), respectively, with the rectangle R replaced by the simplex S. Note that by eliminating one of the variables via $\sum_{j=1}^{n} x_j = 1$, we see that this is a problem in dimension $n - 1$.

5.3 Lower Bounds

The importance of knowing a Lipschitz constant L arises from the following observation. Let P be an n–rectangle or an n–simplex, and let L be a Lipschitz constant of the function f on P, i.e., for all $x, y \in P$, there holds

$$| f(x) - f(y) | \leq L \| x - y \| . \tag{5.11}$$

It follows from (5.11) that, for all $x, y \in P$ we have

$$f(x) \geq f(y) - L \parallel x - y \parallel . \tag{5.12}$$

If $y \in P$ is fixed, then the concave function

$$F(x) = f(y) - L \parallel x - y \parallel \tag{5.13}$$

underestimates $f(x)$ on P.

The following approach for minimizing $f(x)$ over $D \subset \mathbb{R}^n$, $D \subseteq P$, is based directly on (5.13):

Start with an arbitrary point $x_1 \in D$ and define the first approximating function by

$$F_1(x) = f(x_1) - L \parallel x - x_1 \parallel . \tag{5.14}$$

The next iteration point $x_2 \in D$ is defined by

$$F_1(x_2) = \min_{x \in D} F_1(x).$$

Replacing x_1 in (5.14) by x_2 we obtain another concave underestimating function $f(x_2) - L \parallel x - x_2 \parallel$, and the best underestimator of f on D, given the information obtained so far, is

$$F_2(x) = \max_{i=1,2} \{f(x_i) - L \parallel x - x_i \parallel\}. \tag{5.15}$$

Minimizing $F_2(x)$ over D yields the point x_3 satisfying

$$F_2(x_3) = \min_{x \in D} F_2(x).$$

In step k, the approximating function is

$$F_k(x) = \max_{1 \leq i \leq k} \{f(x_i) - L \parallel x - x_i \parallel\}, \tag{5.16}$$

and the next iteration point x_{k+1} is given by

$$F_k(x_{k+1}) = \min_{x \in D} F_k(x). \tag{5.17}$$

It is not difficult to show that every accumulation point of the sequence $\{x_k\}$ is an optimal solution to the problem $\min\{f(x) : x \in D\}$ (Exercise).

The above approach will be referred to as the Piyavskii–Method (see, e.g., Piyavskii (1972)). For *univariate problems*, when $P = D = [a, b] \subset \mathbb{R}$, one usually chooses $x_1 = (a + b)/2$, and obtains $x_2 = a$ or $x_2 = b$, where $x_2 = a$ implies $x_3 = b$ and $x_2 = b$ implies $x_3 = a$. Because of its shape,

the function $F_k(x)$ is often called a *saw-tooth cover* of f over $[a,b]$. Suppose that for $k \geq 3$ the first k evaluation points x_k have already been generated and ordered so that we have

$$a = y_1 \leq y_2 \leq \ldots \leq y_k = b, \text{ where } \{y_1, \ldots, y_k\} = \{x_1, \ldots, x_k\}.$$

The restriction of F_k to the interval $[y_i, y_{i+1}]$ of two consecutive points can be visualized as a *tooth* of the saw–tooth cover on $[y_i, y_{i+1}]$. A straightforward simple calculation shows that the tooth on $[y_i, y_{i+1}]$ is minimized at

$$\tilde{x}_i = (y_i + y_{i+1})/2 + (f(y_i) - f(y_{i+1}))/2L \tag{5.18}$$

with

$$F_k(\tilde{x}_i) = (f(y_i) + f(y_{i+1}))/2 - L(y_{i+1} - y_i)/2 \tag{5.19}$$

(Exercise).

Therefore, (5.17) becomes

$$F_k(x_{k+1}) = \min_{i=1,\ldots,k-1} \{(f(y_i) + f(y_{i+1}))/2 - L(y_{i+1} - y_i)/2\}, \tag{5.20}$$

and $x_{k+1} = \tilde{x}_{i^*}$ for the $i^* \in \{1, \ldots, k-1\}$ where the minimum in (5.20) is attained.

When $P = D$ is a two–dimensional rectangle in $\mathbb{R}^2$, (5.16) and (5.17) can still be evaluated by geometric arguments which take into account the conical shape of $f(x_i) - L \parallel x - x_i \parallel$. For dimensions $n > 2$, however, problem (5.17) constitutes a difficult d.c. optimization problem (which becomes more and more difficult as k increases).

Simpler lower bounds can be derived from (5.13) in the following way. Suppose that the diameter

$$\delta(P) := \sup\{\parallel x - y \parallel : x, y \in P\} \tag{5.21}$$

of P is known. For example, the diameter of an n–rectangle $P = \{x \in \mathbb{R}^n : a \leq x \leq b\}$, is $\delta(P) = \parallel b - a \parallel$, and the diameter of an n–simplex is the length of its longest edge. Let $T \subset P$ denote a *finite* sample of points in P where the function values of f have been evaluated. Then we see from (5.12) that

$$\mu_1(P) = \max_{y \in T} f(y) - L\delta(P) \tag{5.22}$$

is a lower bound for $\inf\{f(x) : x \in P\}$. When P is a rectangle or a simplex the set T often coincides with the vertex set $V(P)$. Note, however, that

it might be worthwhile to take into account "good" interior points. The midpoint $m = (a+b)/2$ of an n–rectangle $a \leq x \leq b$, for example, yields the bound

$$\mu(P) = f(m) - L\delta(P)/2. \tag{5.23}$$

The bound given by

$$\mu_2(P) = \max_{y \in T}\{f(y) - L \max_{\substack{z \in V(P) \\ z \neq y}} \| y - z \|\} \tag{5.24}$$

is more tight but computationally more expensive than (5.22). For $D \subseteq P$, the sharpest lower bound for $\inf\{f(x) : x \in D\}$, given the knowledge of the function values $f(y)$, $y \in T$, and of the Lipschitz constant L, is provided by

$$\min_{x \in D} \max_{y \in T} (f(y) - L \| x - y \|). \tag{5.25}$$

But (5.25) is a difficult optimization problem when the underlying space has dimension $n \geq 2$ (cf. the above discussion of the Piyavskii approach). If $P = S = [v_0, \ldots, v_n]$ is an n–simplex with vertices $v_0, \ldots, v_n$, and $D \subset S$ is a polytope, then (5.25) can be approximated by a *linear program* in the following way. For fixed $y \in T$, the function $F_y(x) = f(y) - L \| x - y \|$ is concave on S, and the uniformly best convex underestimating function $\varphi_y(x)$ of $F_y(x)$ on S (the convex envelope of F_y over S) is known to be that affine function which coincides with $F_y(x)$ at the vertices of S (cf. Section 1.3). For $x \in S$, given by $x = \sum_{i=0}^{n} \lambda_i v_i$, $\sum_{i=0}^{n} \lambda_i = 1$, $\lambda_i \geq 0$ $(i = 0, \ldots, n)$, we have

$$\varphi_y(x) = \sum_{i=0}^{n} \lambda_i F_y(v_i).$$

Replacing $F_y(x) = f(y) - L \| x - y \|$ in (5.25) by its convex envelope $\varphi_y(x)$ we obtain

$$\mu_3(D) = \min_{x \in D} \max_{y \in T} \varphi_y(x). \tag{5.26}$$

Problem (5.26) is equivalent to the linear program

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & \varphi_y(x) \leq t, \; y \in T \\ & x \in D. \end{aligned} \tag{5.27}$$

For $P = D = [v_0, \ldots, v_n]$ and $T = \{v_0, \ldots, v_n\}$, we obtain the following

explicit formulation of (5.27) in the variables $t, \lambda = (\lambda_0, \ldots, \lambda_n)^T$:

$$\begin{array}{ll} \min & t \\ \text{s.t.} & (B, -e)\binom{\lambda}{t} \leq 0, \\ & \lambda \geq 0, \ \sum_{i=0}^{n} \lambda_i = 1, \end{array}$$

where $e = (1, \ldots, 1)^T \in \mathbb{R}^{n+1}$, and B is the $(n+1) \times (n+1)$ matrix with entries $b_{ij} = f(v_i) - L\|v_j - v_i\|$, $0 \leq i, j \leq n$.

It can be shown that μ_3 is a better bound than μ_2, and μ_2 is better than μ_1 (Exercise). To illustrate this, let, for example, $P = D = S = [v_0, \ldots, v_n]$ be an n–simplex and $T = \{v_0, \ldots, v_n\}$. Setting $f_i := f(v_i)$, $0 \leq i \leq n$, and $\delta_{ij} := \|v_i - v_j\|$, $0 \leq i, j \leq n$, we see that

$$\begin{aligned} \mu_1(S) &= \max_{0 \leq k \leq n} f_k - L \max_{0 \leq i < j \leq n} \delta_{ij}, \\ \mu_2(S) &= \max_{0 \leq i \leq n} (f_i - L \max_{0 \leq j \leq n} \delta_{ij}), \\ \mu_3(S) &= \min_{\lambda} \max_{0 \leq i \leq n} (f_i - L \sum_{j=0}^{n} \delta_{ij} \lambda_j) \\ &\quad \text{s.t. } \lambda \geq 0, \ \sum_{j=0}^{n} \lambda_j = 1, \end{aligned}$$

and hence $\mu_1(S) \leq \mu_2(S) \leq \mu_3(S)$.

5.4 Branch and Bound Algorithms

We assume the reader to be familiar with the basic branch and bound algorithm presented in Section 3.7.1 and the corresponding convergence conditions given in Theorem 3.8. We recall that Algorithm 3.5 does not make use of the concavity of the objective function, and that Theorem 3.8 holds for arbitrary continuous functions. We also assume knowledge of the basic partitioning procedure of simplices and rectangles given in Section 3.3.6, and of the notion of an exhaustive subdivision (Definition 3.5, Section 3.7.2). Therefore, those readers who skipped Chapter 3 on concave minimization are requested to go through Section 3.3.6, Section 3.7.1, and the beginning of Section 3.7.2 (until Proposition 3.14) before proceeding to the next section.

5.4.1 Lipschitz Optimization over Rectangles and Simplices

The following algorithm is one of several straightforward realizations of the basic branch and bound method given in Section 3.7.1 for solving the problem

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t.}\ \ a \le x \le b \end{aligned} \tag{5.28}$$

where $a = (a_1, \ldots, a_n)^T$, $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$, $a < b$, and $f : R = \{x : a \le x \le b\} \to \mathbb{R}$ is Lipschitzian on R with Lipschitz constant L.

Algorithm 5.1.

Initialization:

Set $x_R \leftarrow (a+b)/2$, $Q \leftarrow \{a, b, x_R\}$, $\gamma \leftarrow \min\{f(x) : x \in Q\}$;

Choose $v \in Q$ satisfying $f(v) = \gamma$;

Set $\mu(R) \leftarrow \max\{\max\{f(a), f(b)\} - L \parallel b - a \parallel, f(x_R) - L \parallel b - a \parallel /2\}$;

Set $\mu \leftarrow \mu(R)$; $\mathcal{M} \leftarrow \{R\}$; *stop* $\leftarrow$ *false*, $k \leftarrow 1$;

while *stop* = *false* **do**

if $\gamma = \mu$ **then**

stop $\leftarrow$ *true* (v is an optimal solution, and γ the optimal objective function value)

else

Compute $b_j - a_j = \max\{b_i - a_i : i = 1, \ldots, n\}$, and set
$a^1 \leftarrow a$, $b^1 \leftarrow (b_1, \ldots, b_{j-1}, (b_j + a_j)/2, b_{j+1}, \ldots, b_n)^T$,
$a^2 \leftarrow (a_1, \ldots, a_{j-1}, (b_j + a_j)/2, a_{j+1}, \ldots, a_n)^T$, $b^2 \leftarrow b$,
$R_1 \leftarrow \{x : a^1 \le x \le b^1\}$, $R_2 \leftarrow \{x : a^2 \le x \le b^2\}$,
$x_{R_1} \leftarrow (a^1 + b^1)/2$, $x_{R_2} \leftarrow (a^2 + b^2)/2$;

Set

$$\mu(R_1) \leftarrow \max\{\mu(R), \max\{f(a^1), f(b^1)\} - L \parallel b^1 - a^1 \parallel, f(x_{R_1}) - L \parallel b^1 - a^1 \parallel /2\},$$

$$\mu(R_2) \leftarrow \max\{\mu(R), \max\{f(a^2), f(b^2)\} - L \parallel b^2 - a^2 \parallel, f(x_{R_2}) - L \parallel b^2 - a^2 \parallel /2\};$$

Set $Q \leftarrow \{v, b^1, x_{R_1}, a^2, x_{R_2}\}$, $\gamma \leftarrow \min\{f(x) : x \in Q\}$;

Update v satisfying $f(v) = \gamma$;

Set

$$\begin{aligned} \mathcal{M} &\leftarrow (\mathcal{M} \setminus \{R\}) \cup \{R_1, R_2\}, \\ \mathcal{R} &\leftarrow \{R \in \mathcal{M} : \mu(R) < \gamma\}. \\ \mu &\leftarrow \begin{cases} \min\{\mu(R) : R \in \mathcal{R}\}, & \text{if } \mathcal{R} \neq \emptyset \\ \gamma, & \text{if } \mathcal{R} = \emptyset. \end{cases} \end{aligned}$$

Choose

$$R \in \mathcal{R} \text{ satisfying } \mu(R) = \mu;$$

Update a, b such that $R = \{x : a \leq x \leq b\}$.

end if

Set $\mathcal{M} \leftarrow \mathcal{R}$, $k \leftarrow k + 1$.

end while

Algorithm 5.1 uses combinations of the bounds (5.22) (with $T = \{a, b\}$) and (5.23), and bisection of rectangles at the midpoint of one of the longest edges. The first term in the outer max–operation of the bounds $\mu(R_1)$, $\mu(R_2)$ takes into account that the previous bound $\mu(R)$ might happen to exceed the remaining terms.

In a practical implementation one would stop when $\gamma - \mu \leq \varepsilon$ for a given tolerance ε. Moreover, when the diameter of a rectangle becomes small enough one could stop bisecting and try to improve the bounds there by a local nonlinear programming method.

Next we prove convergence of Algorithm 5.1 by showing that the conditions of Theorem 3.8 are fulfilled.

Lemma 5.1. *Successive bisection of n–rectangles at the midpoint of one of its longest edges is exhaustive.*

Proof: Let $R_1 = \{x : a \leq x \leq b\}$ and $\{R_q\}$, $q \geq 1$, be a decreasing sequence of rectangles, where R_{q+1} is obtained from R_q by bisection at the midpoint of one of the longest edges of R_q. Let $c_i = b_i - a_i$ $(i = 1, \ldots, n)$ be ordered such that $c_1 \geq c_i (i = 2, \ldots, n)$. Let $c = (c_1, \ldots, c_1)^T \in \mathbb{R}^n$, and consider the n–cube $C_1 = \{x : a \leq x \leq a + c\}$ and the sequence $\{C_q\}$, $q \geq 1$, of rectangles, where C_{q+1} is obtained from C_q by bisection at the midpoint of one of the longest edges of C_q. Clearly, we have

$$\delta(R_q) \leq \delta(C_q) \quad \forall q \in \mathbb{N}.$$

Moreover, after n bisections, we obtain a cube C_{n+1} with edges of length $c_1/2$, i.e.

$$\delta(R_{n+1}) \leq \delta(C_{n+1}) = \frac{1}{2}\delta(C_1).$$

It follows that $\delta(C_q) \underset{q\to\infty}{\longrightarrow} 0$, and hence $\delta(R_q) \underset{q\to\infty}{\longrightarrow} 0$. □

Let $R_k, Q_k, \mu_k, \gamma_k, v_k$ stand for R, Q, μ, γ, v, respectively at the beginning of iteration k.

Proposition 5.4. *If Algorithm 5.1 is infinite, then*

$$\mu = \lim_{k\to\infty} \mu_k = \lim_{k\to\infty} f(v_k) = \lim_{k\to\infty} \gamma_k = \gamma,$$

and every accumulation point v^* of the sequence $\{v_k\}$ is an optimal solution of $\min\{f(x) : x \in R\}$.

Proof: According to Theorem 3.8 it suffices to show that for every decreasing sequence $R_{k_q}, R_{k_q} \supset R_{k_{q+1}}$, of successively refined rectangles the corresponding bounds satisfy

$$\lim_{q\to\infty} \Big(\gamma_{k_q} - \mu(R_{k_q})\Big) = 0.$$

Let

$$R_{k_q} = \{x : a_{k_q} \leq x \leq b_{k_q}\},\ x_{R_{k_q}} = (b_{k_q} + a_{k_q})/2, \quad \text{and}$$

$$\mu'(R_{k_q}) = \max\{\max\{f(a_{k_q}), f(b_{k_q})\} - L \parallel b_{k_q} - a_{k_q} \parallel,$$
$$f(x_{R_{k_q}}) - L \parallel b_{k_q} - a_{k_q} \parallel /2\}.$$

Moreover, let

$$\gamma(R_{k_q}) = \min\{f(x) : x \in Q_{k_q} \cap R_{k_q}\}.$$

It follows from Lemma 5.1 that $\lim\limits_{q\to\infty} R_{k_q} = \bigcap\limits_{q=1}^{\infty} R_{k_q} = \{s\}$ for some point s, and hence

$$\lim_{q\to\infty} \gamma(R_{k_q}) = \lim_{q\to\infty} \mu'(R_{k_q}) = f(s),$$

since f is continuous and $\delta(R_{k_q}) = \parallel b_{k_q} - a_{k_q} \parallel \underset{q\to\infty}{\longrightarrow} 0$. Using $\gamma(R_{k_q}) \geq \gamma_{k_q}$ and $\mu'(R_{k_q}) \leq \mu(R_{k_q})$, we see that $\gamma_{k_q} - \mu(R_{k_q}) \underset{q\to\infty}{\longrightarrow} 0$ holds. □

Notice that many variants of Algorithm 5.1 are possible, since convergence follows from exhaustiveness of the subdivision and continuity of the

function $f(x)$. Therefore, any exhaustive subdivision of rectangles and any lower bound of the type (5.22) or (5.24) will yield a convergent realization of the basic algorithm given in Section 3.7.1 which can be formulated similarly to Algorithm 5.1. For small dimension n it is certainly worthwhile to consider (5.24). Note, however, that none of the bounds (5.22), (5.23), (5.24) is necessarily monotonically increasing as required in the basic approach in Section 3.7.1. Therefore, whenever a rectangle R_i is obtained from a rectangle R by direct subdivision we have to ensure monotonicity by setting $\mu(R_i) = \max\{\mu(R), \mu'(R_i)\}$, where $\mu'(R_i)$ is one of the bounds in (5.22) – (5.24).

In a similar way, various branch and bound algorithms can be formulated to minimize a Lipschitz function over an n–simplex S. Any exhaustive subdivision combined with one of the lower bounds in (5.22) – (5.24) or (5.26), (5.27) with $D = S$ yields a convergent realization of the basic Algorithm 3.5 in Section 3.7.1. A detailed formulation similarly to Algorithm 5.1 is left to the reader as an exercise.

Example 5.4. We continue the discussion of Section 5.2.2. on solving systems of inequalities.
Let, for $i = 1, \ldots, m$, the functions $f_i : S \to \mathbb{R}$ be Lipschitz functions on a simplex $S = [v_0, \ldots, v_n] \subset \mathbb{R}^n$ with Lipschitz constants L_i. We want to find a solution $x^* \in S$ of the system of inequalities $f_i(x) \leq 0$ $(i = 1, \ldots, m)$ (or detect that such a solution does not exist in S). For industrial applications of this problem, see, e.g., Hendrix and Pintér (1991). The discussion in Section 5.2.2. suggests to apply a simplicial modification of Algorithm 5.1 to the Lipschitz optimization problem

$$\min_{x \in S} \max_{i=1,\ldots,m} f_i(x)$$

until a point $x^* \in S$ satisfying $\max_{i=1,\ldots,m} f_i(x^*) \leq 0$ is found (or it is shown that $\max_{i=1,\ldots,m} f_i(x) > 0$ for all $x \in S$). The objective function $\max_{i=1,\ldots,m} f_i(x)$ has the Lipschitz constant $L = \max_{i=1,\ldots,m} L_i$, and we could use one of the bounds $\mu_k = \mu_k(S)$ $(k = 1, 2, 3)$ above with this Lipschitz constant and, e.g., $T = V(S)$. Better bounds can be found by considering each function f_i separately. For example, in the case of μ_1, the number

$$\tilde{\mu}_1 := \max_{i=1,\ldots,m} \mu_{1i},$$

where

$$\mu_{1i} = \max_{v \in V(S)} f_i(v) - L_i \delta(S) \quad (1 \leq i \leq m),$$

yields a lower bound. This follows from

$$\mu_{1i} \leq \min_{x \in S} f_i(x) \quad (1 \leq i \leq m),$$

and hence

$$\tilde{\mu}_1 \leq \max_{i=1,\ldots,m} \min_{x \in S} f_i(x) \leq \min_{x \in S} \max_{i=1,\ldots,m} f_i(x).$$

It is easy to see that

$$\tilde{\mu}_1 = \max_{i=1,\ldots,m} (\max_{v \in V(S)} f_i(v) - L_i \delta(S)) \geq \max_{v \in V(S)} \max_{i=1,\ldots,m} f_i(v) - \max_{i=1,\ldots,m} L_i \delta(S) = \mu_1.$$

Similar arguments hold for the bounds $\tilde{\mu}_k = \max_{i=1,\ldots,m} \mu_{ki}$, $k = 2, 3$, where μ_{ki} is the corresponding bound of the function f_i. In the case of

$$\tilde{\mu}_3 = \max_{1 \leq i \leq m} \mu_{3i} = \max_{1 \leq i \leq m} \min_{\lambda} \max_{0 \leq \ell \leq n} (\varphi_\ell^i(\lambda) : \lambda \geq 0, e^T \lambda = 1),$$

where

$$\varphi_\ell^i(\lambda) = f_i(v_\ell) - L_i \sum_{j=0}^{n} \delta_{ij} \lambda_j,$$

one has to solve m linear programming problems. This bound, however, can be improved to

$$\tilde{\mu}_4 = \min_{\lambda} \max_{1 \leq i \leq m} \max_{0 \leq \ell \leq n} (\varphi_\ell^i(\lambda) : \lambda \geq 0, e^T \lambda = 1\} \geq \tilde{\mu}_3,$$

which requires solving only a single linear program of the form

$$\min_{\lambda, t} \{t : \varphi_\ell^i \leq t \ (1 \leq i \leq m, \ 0 \leq \ell \leq n), \ \lambda \geq 0, \ e^T \lambda = 1, \ t \in \mathbb{R}\}.$$

The following small example illustrates the quality of the different bounds: a simplicial variant of Algorithm 5.1 which uses successive bisection was applied for solving the problem

$$\min_{x \in S} f(x),$$

where $S = \{x \in \mathbb{R}^3 : x \geq 0, \ \sum_{i=1}^{3} x_i = 1\}$ and

$$\begin{aligned} f(x) &= \max\{f_1(x), f_2(x)\}, \\ f_1(x) &= -1.0 + 8x_1 + 8x_2 - 32x_1x_2, \\ f_2(x) &= 3.6 - 12x_1 - 4x_3 + 4x_1x_3 + 10x_1^2 + 2x_3^2. \end{aligned}$$

For each simplex, the Lipschitz constants L_1 and L_2 were determined by maximizing $\| \nabla f_1 \|$ and $\| \nabla f_2 \|$, respectively, over its vertex set. The algorithm was terminated when the difference between upper and lower bound became less than 0.01. The following table shows the number of iterations (Iter) and the maximal number of partition sets (mps) for the different types of bounds:

Bound	Iter	mps
μ_1	6032	1332
μ_2	4174	1020
μ_3	2145	618
$\tilde{\mu}_1$	508	84
$\tilde{\mu}_2$	413	69
$\tilde{\mu}_3$	267	56
$\tilde{\mu}_4$	221	47

5.4.2 Linear Constraints

Let D be an n–dimensional polytope in $\mathbb{R}^n$ and assume that $f(x)$ is Lipschitzian with Lipschitz constant L on an n–simplex $S_o \supset D$ with known vertex set $V(S_o)$ which tightly approximates D. Several ways for constructing such a simplex are discussed in the beginning of Section 3.5.2. Consider the global optimization problem

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x \in D. \end{aligned} \tag{5.29}$$

Simplicial branch and bound technique

We apply Algorithm 3.5 (Section 3.7.1) with the following specifications:

- The partition sets are n–simplices S with known vertex set $V(S)$ generated from S_o by successive bisection at the midpoint of one of the longest edges (recall from Section 3.7.2 that this subdivision is exhaustive).
- For each simplex $S = [v_o, \ldots, v_n]$ with vertex set $V(S) = \{v_o, \ldots, v_n\}$ a lower bound
$$\mu'(S) \leq \min\{f(x) : x \in S \cap D\}$$
is obtained as follows:

Set

$$T = V(S) \cup \{b\},$$

where

$$b = \frac{1}{n+1} \sum_{i=0}^{n} v_i,$$

and, for each $y \in T$, let $\varphi_y(x)$ be the convex envelope of $F_y(x) = f(y) - L \parallel x - y \parallel$ over S ($\varphi_y(x)$ is an affine function, cf. Section 5.3). Solve the *linear program*

$$\begin{array}{ll} \min & t \\ \text{s.t.} & \varphi_y(x) \leq t,\ y \in T, \\ & x \in D \cap S \end{array} \tag{5.30}$$

and set $\mu'(S) = t^*$, the optimal value of this program. When $D \cap S = \emptyset$ (which is checked, for example, by Phase I of the simplex algorithm for solving (5.30)), the partition set is eliminated. Otherwise, suppose that S was obtained from a simplex S' by bisection of S'. Then set

$$\mu(S) = \max\{\mu(S'), \mu'(S)\} \tag{5.31}$$

in order to guarantee monotonicity of the lower bounds $\mu(S)$.

- Feasible points in D are computed while solving the linear program (5.30) so that we obtain nonempty sets Q for determining upper bounds as required in Algorithm 3.5.

A detailed formulation of the resulting algorithm is straightforward and left to the reader as an exercise.

Convergence in the sense of Theorem 3.8 is a consequence of the following simple argument. For each simplex S replace in the above technique the bound $\mu'(S)$ by $\overline{\mu}'(S) = f(y) - L\delta(S)$, where y is an arbitrary vertex of S and $\delta(S)$ is the diameter of S. Moreover, suppose that each simplex S satisfying $S \cap D = \emptyset$ is eliminated. By exactly the same reasoning as in Proposition 5.4, we then see that – in view of exhaustiveness of the subdivision and continuity of $f(x)$ – the assertion of Proposition 5.4 holds similarly for the modified algorithm. Formally $\min\{f(x) : x \in R\}$ in Proposition 5.4 has to be replaced by $\min\{f(x) : x \in D\}$.

The corresponding convergence result for the above technique follows from the obvious fact that the bound $\mu'(S)$ obtained from (5.30) satisfies $\mu'(S) \geq \overline{\mu}'(S)$.

5.4.3 Lipschitz Constraints

We briefly discuss an approach for solving constrained Lipschitzian optimization problems which has been called "deletion-by-infeasibility method". This approach does not require linearity of the constraints. Therefore, we consider the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0 \end{aligned} \tag{5.32}$$

where

$$g(x) = \max_{i=1,\ldots,m} g_i(x), \tag{5.33}$$

and f, g are Lipschitzian on an n–simplex S_o containing the feasible set $D = \{x \in \mathbb{R}^n : g(x) \leq 0\}$. We assume that $\text{int}D \neq \emptyset$ and $D = cl(\text{int}D)$ (the closure of $\text{int}D$) (i.e., D is a robust set).

Now, let us ignore for a moment the constraint $g(x) \leq 0$, and consider the problem of finding $\min\{f(x) : x \in S_o\}$. If we minimize $f(x)$ over S_o by one of the exhaustive simplicial branch and bound techniques discussed in Section 5.4.1, and the limit v^* of one of the subsequences of the sequence $\{v_k\}$ happens to be in D, then we are done, since v^* must be optimal for (5.32) because of $\min\{f(x) : x \in S_o\} \leq \min\{f(x) : x \in D\}$. Of course, we cannot expect that $v^* \in D$, since $\min\{f(x) : x \in S_o\}$ can be attained in $S_o \setminus D$. But if we were able to detect and delete each simplex S generated by such an algorithm for finding $\min\{f(x) : x \in S_o\}$ which satisfies $S \cap D = \emptyset$, then every decreasing sequence $\{S_q\}$ of remaining simplices would eventually shrink to a point in D.

As a consequence of exhaustiveness of the subdivision and continuity of the function f the sequence $\{\mu_k\}$ of lower bounds would converge to the optimal value of (5.32) for each of the bounding rules (5.22) – (5.24) or (5.26), (5.27). Notice, however, that the upper bounds γ_k defined as best objective function values at feasible points would not necessarily be available.

A simplex S is called *infeasible* if $S \cap D = \emptyset$, otherwise it is called *feasible.*

A straightforward implementation of the above ideas turns out to be very expensive computationally. To see this, let S be a simplex with known vertex set $V(S)$. If $g(v) \leq 0$ for some $v \in V(S)$, then S is obviously feasible.
If $g(v) > 0$ for each $v \in V(S)$, then, however, the simplex can be feasible or infeasible. In this case, in order to decide whether $S \cap D = \emptyset$ or not, we

could consider the optimization problem

$$\begin{aligned} &\min\ g(x) \\ &\text{s.t.}\ \ x \in S. \end{aligned} \tag{5.34}$$

Clearly, S is infeasible if and only if

$$g^* = \min\{g(x) : x \in S\} > 0. \tag{5.35}$$

But (5.35) is a Lipschitz optimization problem which we certainly do not want to solve for each partition set whose feasibility is not known. Therefore, we propose the following approximate *deletion-by-infeasibility* rule (DR) which is based on the bounds (5.22) – (5.27):

(DR): *Let $L(g,S)$ be a Lipschitz constant of g over S, and let T be a finite set of points in S. Delete S whenever one of the following quantities $G_i(S)$, $i \in \{1,2,3\}$ is positive:*

(i)

$$G_1(S) = \max_{y \in T} g(y) - L(g,S)\delta(S). \tag{5.36}$$

(ii)

$$G_2(S) = \max_{y \in T}\{g(y) - L(g,S) \max_{\substack{z \in T \\ z \neq y}} \| y - z \|\}. \tag{5.37}$$

(iii)

$$G_3(S) = \min_{x \in S} \max_{y \in T} \varphi_y(x), \tag{5.38}$$

where $\varphi_y(x)$ is the convex envelope of $g(y) - L \| y - x \|$ over S.

Each of the quantities G_i, $i \in \{1,2,3\}$ constitutes a lower bound of $g(x)$ over S (recall that (5.36) corresponds to (5.22), (5.37) to (5.24), and (5.38) to (5.26), where (5.38) can be obtained by solving a linear program).

Therefore, a partition set which is deleted according to the rule (DR) must be infeasible. It is clear, however, that infeasible partition sets can occur which are not eliminated.

The rule (DR) is "certain" only in the limit:

Lemma 5.2. *Let $\{S_q\}$ be a decreasing sequence of simplices generated by an exhaustive subdivision process of a branch and bound procedure which deletes partition sets according to the rule (DR). Then the limit point $\{s\} = \lim_{q \to \infty} S_q = \bigcap_{q=1}^{\infty} S_q$ satisfies $g(s) \leq 0$.*

Proof: We may assume that there is a bound L_o satisfying

$$L_o \geq L(g, S_q) \text{ for all } q$$

(notice that we certainly have $L(g, S_q) \leq L(g, S_o)$).

Suppose that we apply the rule (DR) with $G_1(S_q)$. Then, we have, for every q,

$$0 \geq G_1(S_q) = \max_{y \in T_q} g(y) - L(g, S_q)\delta(S_q) \geq \max_{y \in T_q} g(y) - L_o\delta(S_q),$$

and hence, by continuity of g and $\delta(S_q) \underset{q \to \infty}{\longrightarrow} 0$, $T_q \subset S_q$,

$$0 \geq \lim_{q \to \infty} G_1(S_q) = g(s).$$

The proof is similar when $G_2(S_q)$ or $G_3(S_q)$ is used for every q. □

In order to solve

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in D \end{aligned} \tag{5.39}$$

where $D := \{x : g(x) \leq 0\} \subset S_o$ we can now use a branch and bound algorithm for solving

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in S_o. \end{aligned}$$

Suppose that the subdivision is exhaustive and that one of the bounds in (5.22) – (5.27) is applied throughout. The modification consists in the incorporation of the deletion rule (DR) with one of the functions $G_i(S)$, $i \in \{1, 2, 3\}$. Moreover, we do not consider upper bounds γ for $\min\{f(x) : x \in D\}$ since these are not guaranteed to be available.

To each lower bound $\mu(S)$ we associate a point $\overline{x} \in S$ where $\mu(S)$ is attained. When (5.22) is used, we define $\overline{x}$ by

$$f(\overline{x}) - L\delta(S) = \max_{y \in T} f(y) - L\delta(S).$$

In the case (5.24) the point $\overline{x}$ satisfies

$$f(\overline{x}) - L \max_{\substack{z \in V(P) \\ z \neq \overline{x}}} \| \overline{x} - z \| = \max_{y \in T} \{f(y) - L \max_{\substack{z \in V(P) \\ z \neq y}} \| y - z \|\},$$

whereas the bound (5.26), (5.27) yields a point $\overline{x}$ where

$$\min_{x \in S} \max_{y \in T} \varphi_y(x)$$

is attained.

As usual, we associate the index k to each quantity at the beginning of iteration k.

Proposition 5.5. *Each modified branch and bound algorithm described above satisfies*

$$\lim_{k\to\infty} \mu_k = \min\{f(x) : x \in D\},$$

and every accumulation point $\overline{x}^*$ of the sequence $\{\overline{x}_k\}$ is an optimal solution to Problem (5.39).

Proof: Let $\overline{x}^*$ be an accumulation point of $\{\overline{x}_k\}$ satisfying $\lim_{q\to\infty} \overline{x}_{k_q} = \overline{x}^*$. Moreover, let $\{S_{k_q}\}$ denote the corresponding sequence of simplices such that for all q, $\overline{x}_{k_q} \in S_{k_q}$ and $\mu_{k_q} = \mu(S_{k_q})$. As we have seen while discussing Algorithm 3.5 we can assume that $\{S_{k_q}\}$ is monotonically decreasing, i.e., $S_{k_{q+1}} \subset S_{k_q}$ for every q. Exhaustiveness then implies $\lim_{q\to\infty} S_{k_q} = \{\overline{x}^*\}$. It follows from Lemma 5.2 that $\overline{x}^* \in D$.

Furthermore, by similar arguments as in the proof of Proposition 5.4, we see from exhaustiveness, from the way how the bounds $\mu(S_{k_q})$ are defined, and from continuity of the objective function, that

$$\lim_{q\to\infty} \mu_{k_q} = \lim_{q\to\infty} \mu(S_{k_q}) = f(\overline{x}^*). \tag{5.40}$$

By construction, the sequence $\{\mu_k\}$ is nondecreasing and bounded from above by $f^* = \min\{f(x) : x \in D\}$. Therefore, $\mu = \lim_{k\to\infty} \mu_k \leq f^*$ exists, and (5.40) tells us that

$$\lim_{k\to\infty} \mu_k = f(\overline{x}^*) \leq f^*.$$

But $\overline{x}^* \in D$ implies $f(\overline{x}^*) \geq f^*$, which completes the proof. □

Remarks:

(i) Rectangular branch and bound algorithms can be derived in a similar way when D is contained in an n–rectangle R. In this case, bounds of the type (5.22) – (5.24) and the deletion mechanism corresponding to G_1 and G_2 can be applied. Rectangular algorithms may be more efficient regarding storage requirements since a rectangle $R = \{x : a \leq x \leq b\}$ is obviously defined by the two points a and b.

(ii) Different "deletion–by–infeasibility rules" can be derived for the following two cases:

α) $D = \{x : g(x) \leq 0\}$ with $g : \mathbb{R}^n \to \mathbb{R}$ convex,

and

β) the feasible set is the intersection of a convex set and a finite number of complements of convex sets (cf. Horst and Tuy (1993); see also Exercise 5.11).

5.5 Implementation of Branch and Bound Methods and Numerical Results

5.5.1 General Remarks

In a branch and bound procedure the elements of the current partition have to be stored together with the values of its lower bounds. In the corresponding array, items with minimum value of the lower bounds have to be selected. Certain items must be deleted, and new items will be inserted.
The length ℓ of these arrays can become very large such that the effect of an efficient performance of these operations may surpass any other algorithmic effort. This observation holds in particular when rather crude bounds are used and (or) high precision is required. Therefore, an important point in the implementation of any branch and bound algorithm is the choice of an appropriate data structure which guarantees fast execution of the above operations. Doubly linked lists, for example, require $O(\ell)$ time for each operation. A number of more sophisticated data structures, however, take only $O(\log \ell)$ time in the worst case. Presentation of details is beyond our scope here. These can be found in various excellent computer science textbooks. Notice that bisection of partition sets leads to binary trees which are very well studied in the literature. Apart from the common management tasks, many branch and bound algorithms involve common subproblems such as, for example, bisection of partition elements, γ–extension (cf. Chapters 2 and 3), and linear programs. Therefore, in order to develop a library of different branch and bound programs, structured top–down refinement and modularity with parameter dependent modules is at least as important as in any other software developments. Memory space can be crucial and we have seen above that the length ℓ of the array should be kept to a minimum. Moreover, the number of data required for storing a single partition element depends on the dimension of the variable space. Therefore, it is clear that a language with call–by–value concept and dynamic memory allocation at the data level should be used. Most modern programming languages meet these demands.

Until the early 1990's, most computer codes have been written in the classical numerical analysis language FORTRAN77. Transition to more flexible languages such as one of the C versions of FORTRAN90 is being observed subsequently.

In branch and bound applications one often detects a good feasible point very early such that most of the computational effort is used to verify its quality or prove optimality. Typically, when high precision is required, large parts of the feasible domain are deleted fast while the procedure takes many iterations to create nested little partition elements in a neighborhood of an optimal solution. Various heuristics and problem dependent devices have been proposed, such as deletion of sufficiently small partition elements, combination of global optimization tools with standard nonlinear local optimization, decomposition schemes etc.

5.5.2 Numerical Example

The simplicial algorithm discussed in Section 5.4.2 has been implemented in C++ and was run for most of the testproblems known from the literature. For a survey of these test problems and additional numerical tests, see Hansen and Jaumard (1994).

The following class of examples turns out to be rich enough to produce typical numerical results. We seek an ε–optimal solution (in the sense described above) of

$$\min f(x) = -\sum_{i=1}^{m} 1/(\|x - a_i\|^2 + c_i) \tag{5.41}$$

$$\text{s.t.} \quad x \in S = \{x \in \mathbb{R}^n_+, \ \sum_{j=1}^{n} x_j \leq 10n\}, \tag{5.42}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ with different values of the parameters $m \in \mathbb{N}$, and, for $1 \leq i \leq m$,

$$a_i \in \{x \in \mathbb{R}^n : 0 \leq x_j \leq 10, \ 1 \leq j \leq n\}, \ c_i > 0. \tag{5.43}$$

For a given simplex $S = [v_0, \ldots, v_n]$, a Lipschitz constant

$$L_f(S) \geq \max\{\|\nabla f(x)\| : x \in S\} \tag{5.44}$$

can be derived as follows. We have

$$f(x) = \sum_{i=1}^{m} f_i(x), \ f_i(x) := -1/[(x - a_i)^T(x - a_i) + c_i], \tag{5.45}$$

and hence

$$\left|\frac{\partial f_i}{\partial x_k}\right| \leq |2(x_k - a_{i,k})|((x_k - a_{i,k}^2 + c_i)^2|, \tag{5.46}$$

where $a_{i,k}$ denotes the k–th component of a_i. The function $h : \mathbb{R} \to \mathbb{R}$ defined by

$$h(t) := 2|t - a| \cdot ((t - a)^2 + c)^{-2}$$

attains its maximum at the points $t_{1,2} = a \pm \sqrt{c/3}$, where

$$h(t_i) = (3/8)\sqrt{3}c^{-3/2},\ i = 1, 2. \tag{5.47}$$

Therefore,

$$||\nabla f(x)|| \leq (3/8)\sqrt{3n}\sum_{i=1}^{m} c_i^{-3/2} \tag{5.48}$$

on $\mathbb{R}^n$. It is easy to see that the bound (5.48) can be tightened on a simplex $S = [v_0, \ldots, v_n]$ to

$$\max\{||\nabla f(x)|| : x \in S\} \leq L_f(x) := \sqrt{\sum_{k=1}^{n}(\sum_{i=1}^{m} \alpha_{ik})^2}, \tag{5.49}$$

where

$$\alpha_{ik} := \begin{cases} (3/8) \cdot \sqrt{3} \cdot c_i^{-1.5} & \text{if } \{t_{i,k}^1, t_{i,k}^2\} \cap [\ell_k, u_k] \neq \emptyset \\ \max\{g_{ik}(l_k), g_{ik}(u_k)\} & \text{else,} \end{cases} \tag{5.50}$$

with

$$g_{i,k}(t) = |2(t - a_{i,k})|((t - a_{i,k})^2 + c_i)^{-2},\ t_{i,k}^{1,2} = a_{i,k} \pm \sqrt{c_i/3}, \tag{5.51}$$

and

$$\ell_k = \min\{v_{i,k} : 0 \leq i \leq n\},\ u_k = \max\{v_{i,k} : 0 \leq i \leq n\}. \tag{5.52}$$

For each simplex S, the Lipschitz constant given in (5.49) – (5.52) was used. For the numerical tests, we chose $n = 2$, and considered the four objective functions

$$f_j(x) = -\sum_{i=1}^{m_j} 1/(||x - a_i||^2 + c_i),\ 1 \leq j \leq 4, \tag{5.53}$$

where $m_1 = 2$, $m_2 = 5$, $m_3 = 8$, $m_4 = 10$, and the parameters c_i, $a_{i,k}$ from Table 5.1.

i	1	2	3	4	5	6	7	8	9	10
c_i	0.70	0.73	0.76	0.79	0.82	0.85	0.88	0.91	0.94	0.97
$a_{i,1}$	4.0	2.5	7.5	8.0	2.0	2.0	4.5	8.0	9.5	5.0
$a_{i,2}$	4.0	3.8	5.6	8.0	1.0	8.5	9.5	1.0	3.7	0.3

Table 5.1: Parameters for the numerical examples

Figure 5.1 shows a plot of the function $-f_4(x)$ over $[0,10] \times [0,10]$.

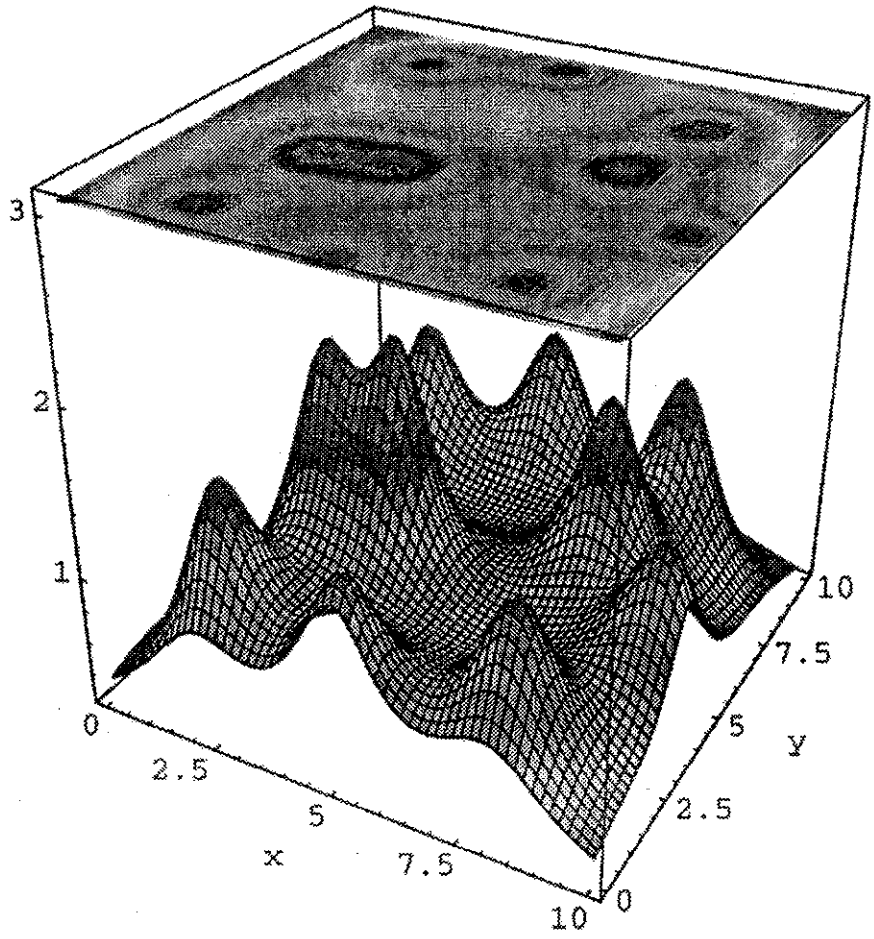


Figure 5.1: Plot of the function $-f_4(x)$

The following Table 5.2 displays relevant results on the computational effort required while minimizing the function f_i, $1 \leq i \leq 4$ over the initial simplex given in (5.42) by means of our branch and bound method with the lower bounds (*Lbd*) μ_1, μ_2, μ_3 and the four precisions $\varepsilon_k = 10^{-k}$, $1 \leq k \leq 4$. The sets T and Q are set to $V(S)$, bisection was used throughout. The column *Iter* contains the required number of iterations. (Notice that

each iteration calls for one function evaluation and two evaluations of the Lipschitz constant (5.49) – (5.52). The column *mps* shows the maximal number of partition sets to be stored and administrated in one iteration. The abbreviations T_{avl} and T_{dll} stand for computing time (in seconds) when the partition sets were administrated by AVL–trees (which require $O(\log \ell)$ times for the basic administrative operations) or doubly linked linear lists (which require ($O(\ell)$–time), respectively (cf., e.g., Kruse et.al. (1991)). The interpretation of the results is obvious.

		$f_1(x)$				$f_2(x)$			
Lbd	ε	**Iter**	**mps**	$\mathbf{T_{avl}}$	$\mathbf{T_{dll}}$	**Iter**	**mps**	$\mathbf{T_{avl}}$	$\mathbf{T_{dll}}$
μ_1	ε_1	1380	470	0.57	0.62	3025	989	1.52	2.22
μ_2	ε_1	881	283	0.37	0.38	1875	604	0.97	1.03
μ_3	ε_1	543	182	0.55	0.52	1185	397	1.27	1.27
μ_1	ε_2	2930	532	1.08	1.18	6189	1138	3.02	5.07
μ_2	ε_2	1882	335	0.77	0.68	3731	670	1.87	2.12
μ_3	ε_2	1141	208	1.07	1.00	2340	464	2.45	2.47
μ_1	ε_3	12272	3477	4.68	17.35	25138	7036	12.63	76.52
μ_2	ε_3	6934	1855	2.63	5.77	13804	3711	6.97	22.52
μ_3	ε_3	4107	1128	3.78	4.62	8228	2255	8.58	13.38
μ_1	ε_4	103438	34623	43.83	1959.47	212233	70835	117.22	8876.58
μ_2	ε_4	53701	17630	22.65	484.50	109031	36008	62.12	2161.18
μ_3	ε_4	32252	10420	31.02	176.00	65362	21209	71.62	750.68

		$f_3(x)$				$f_4(x)$			
Lbd	ε	**Iter**	**mps**	$\mathbf{T_{avl}}$	$\mathbf{T_{dll}}$	**Iter**	**mps**	$\mathbf{T_{avl}}$	$\mathbf{T_{dll}}$
μ_1	ε_1	5661	1923	3.53	7.02	7719	2698	5.43	12.27
μ_2	ε_1	3358	1105	2.12	2.97	4557	1533	3.23	5.15
μ_3	ε_1	2145	761	2.60	2.90	2874	1016	3.72	4.33
μ_1	ε_2	12300	2189	7.62	17.60	17764	3322	12.38	35.33
μ_2	ε_2	7194	1260	4.53	7.37	10116	1849	7.18	13.37
μ_3	ε_2	4417	867	5.18	6.28	6238	1151	7.83	9.83
μ_1	ε_3	57636	17646	37.73	477.77	96084	29751	69.28	1500.02
μ_2	ε_3	30636	9085	20.32	121.90	50327	15239	35.77	369.47
μ_3	ε_3	18795	5431	22.63	55.37	29844	9225	37.55	139.02
μ_1	ε_4	521664	173774	350.20	58474	885905	296556	657.22	174357
μ_2	ε_4	265345	87728	176.72	14342	448745	149449	330.82	42364
μ_3	ε_4	156377	53385	190.00	4783	267643	89294	344.97	14657

Table 5.2: Computational Results

5.6 Exercises

5.1 Let f be a Lipschitz function defined on an interval $[a,b]$, with Lipschitz constant L. Show the following: if there exist $x_1, x_2 \in [a,b]$, $x_1 < x_2$ such that $f(x_1) = f(x_2) + L(x_2 - x_1)$ (respectively $f(x_1) = f(x_2) - L(x_2 - x_1)$), then $f(x) = f(x_2) + L(x_2 - x)$ (respectively $f(x) = f(x_2) - L(x_2 - x)$) for all $x \in [x_1, x_2]$).

5.2 Show that for $1 \leq p \leq \infty$ and $z \in \mathbb{R}^n$

$$\| z \|_p \leq \| z \|_1 .$$

5.9

5.3 Verify the Lipschitz constants in Example 5.1 and Example 5.2.

5.10

5.4 Complete the proof of Proposition 5.2.

5.5 Let $f_i : R \to \mathbb{R}$ be Lipschitzian on the rectangle $R = \{x \in \mathbb{R}^n : a \leq x \leq b\}$, $a, b \in \mathbb{R}^n$, $a < b$, with Lipschitz constants L_i $(i = 1, \ldots, m)$. Formulate a Lipschitz optimization problem which allows one to compute a solution $x^* \in R$ of the system

5.11

$$f_i(x) = 0 \ (i = 1, \ldots, p), \ f_i(x) \leq 0 \ (i = p+1, \ldots, m).$$

5.6 Let $f : [a,b] \to \mathbb{R}$ be Lipschitzian on the real interval $[a,b]$ with Lipschitz constant L. Show that every accumulation point of the sequence $\{x_k\}$ generated by the Algorithm of Piyavskii solves the problem $\min\{f(x) : x \in [a,b]\}$.

5.7 Verify formulas (5.18) and (5.19).

5.8 Consider the univariate optimization problem to determine $f^* = \min\{f(x) : x \in [a,b]\}$, $a, b \in \mathbb{R}$, $a < b$, where $f : [a,b] \to \mathbb{R}$ is Lipschitzian on $[a,b]$ with Lipschitz constant L. Given k different evaluation points $x_i \in [a,b]$ $(i = 1, \ldots, k)$, the best underestimating function which we can obtain from knowledge of the function values $f(x_i)$ $(i = 1, \ldots, k)$ and of L is given by the "saw–tooth cover"

$$F(x) = \max_{i=1,\ldots,k} \{f(x_i) - L \mid x - x_i \mid\}. \qquad (*)$$

Let $\varepsilon > 0$ be a small tolerance and try to find a point $x_\varepsilon^* \in [a,b]$ satisfying $f(x_\varepsilon^*) \leq f^* + \varepsilon$.

a) Show that x_ε^* can be found by k function evaluations, where k is the smallest integer satisfying $k \geq \frac{L(b-a)}{2\varepsilon}$.

b) Assume that f^* is known. Construct a "reference saw–tooth cover" of the type $(*)$ which has all its local minimum values (downward peaks) in the interior of $[a,b]$ at $f^* - \varepsilon$.

c)* Let $k_B(\varepsilon)$ denote the number of evaluation points required by the reference saw–tooth cover in b). Show that, for any $\varepsilon > 0$, the number of evaluation points $k_P(\varepsilon)$ required by the Piyavskii–Method to obtain x_ε^* satisfies

$$k_P(\varepsilon) \leq 4k_B(\varepsilon) + 1.$$

5.9 Show the equivalence of problem (5.26) and problem (5.27).

5.10 Formulate a convergent branch and bound algorithm for minimizing a Lipschitz function over an n–simplex by using bisection at the midpoint of one of the longest edges and the lower bounds (5.26), (5.27). Prove its convergence.

5.11 Consider the simplicial branch and bound approach of Section 5.4.3 for minimizing a Lipschitz function $f(x)$ over $D = \{x : g(x) \leq 0\}$. Let now $g : \mathbb{R}^n \to \mathbb{R}$ be convex and D compact, with known $y \in \text{int}D$. For each simplex S satisfying $V(S) \cap D = \emptyset$, choose $v \in V(S)$ and let H be the hyperplane which supports D at the intersection of the line segment $[y,v]$ with the boundary of D. Delete S whenever

$$V(S) \subset \overset{\circ}{H}_+, \qquad (*)$$

where $\overset{\circ}{H}_+$ is the open halfspace generated by H satisfying $y \notin \overset{\circ}{H}_+$.

(i) Give a formula for $\overset{\circ}{H}_+$ and indicate how the quantities involved in that formula can be computed when $g(x) = \max_{i=1,\ldots,m} g_i(x)$ with $g_i : \mathbb{R}^n \to \mathbb{R}$ convex and differentiable.

(ii) Prove that Lemma 5.2 holds when the deletion rule (DR) is replaced by $(*)$.