Preface

Books on a technical topic – like linear programming – without exercises ignore the principal beneficiary of the endeavor of writing a book, namely the student – who learns best by doing exercises, of course. Books with exercises – if they are challenging or at least to some extent so – need a solutions manual so that students can have recourse to it when they need it. Here we give solutions to all exercises and case studies of M. Padberg’s Linear Optimization and Extensions (second edition, Springer-Verlag, Berlin, 1999). In addition we have included several new exercises and taken the opportunity to correct and change some of the exercises of the book. Here and in the main text of the present volume the terms “book”, “text” etc. designate the second edition of Padberg’s LP book and the page and formula references refer to that edition as well. All new and changed exercises are marked by a star * in this volume. The changes that we have made in the original exercises are inconsequential for the main part of the original text where several of the exercises (especially in Chapter 9) are used on several occasions in the proof arguments. None of the exercises that are used in the estimations, etc. have been changed. Quite a few exercises instruct the students to write a program in a computer language of their own choice. We have chosen to do that in most cases in MATLAB without any regard to efficiency, etc. Our prime goal here is to use a macro-language that resembles as closely as possible the mathematical statement of the respective algorithms. Once students master this first level, they can then go ahead and discover the pleasures and challenges of writing efficient computer code on their own.

To make the present volume as self-contained as possible, we have provided here summaries of each chapter of Padberg’s LP book. While there is some overlap with the text, we think that this is tolerable. The summaries are –in almost all cases– without proofs, thus they provide a “mini-version” of the material treated in the text. Indeed, we think that having such summaries without the sometimes burdensome proofs is an advantage to the reader who wants to acquaint herself/himself with the material treated at length in the text. To make the cross-referencing with the text easy for the reader, we have numbered all chapters (and most sections and subsections) as well as the formulas in these summaries exactly like in the text. Moreover, we have reproduced here most of the illustrations of the text as we find these visual aids very helpful in communicating the material. Finally, we have reproduced here the appendices of the text as the descriptions of the cases contained therein would have taken too much space anyway.

We have worked on the production of this volume over several years and did so quite frequently at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) in Berlin, Germany, where Alevras was a research fellow during some of this time. We are most grateful to ZIB’s vice-president, Prof. Dr. Martin Grötschel, for his hospitality and tangible support of our endeavor. Padberg’s work was also supported in part through an ONR grant and he would like to thank Dr. Donald Wagner of the Office of Naval Research, Arlington, VA, for his continued support.

New York City, January, 2001

Dimitris Alevras
Manfred Padberg
9. Ellipsoid Algorithms

Divide et impera!1
Niccolo Machiavelli (1469-1527 A.D.)

Here we summarize the essentials of Chapter 9 of the text. We consider the linear optimization problem over a rational polyhedron \( P \subseteq \mathbb{R}^n \) of facet complexity \( \phi \)
\[
\max \{ cx : x \in P \}.
\]

In Chapter 7.5.3 we reduced the problem of polynomial solvability of this problem to the question of the existence of subroutines \( \text{FINDXZ}(P, n, \phi, \Phi, c, z, x, \text{FEAS}) \) or \( \text{FINDXZ}(P, n, \phi, c, z, x, \text{FEAS}) \) that solve a feasibility problem in polynomial time. The ellipsoid algorithm settles this existence question in a theoretically satisfactory way for any rational polyhedron \( P \subseteq \mathbb{R}^n \).

By point 7.5(d), we can replace \( P \) by a rational polytope \( P_\Phi \) of equal dimension without changing the optimization problem. We assume that we have a linear description \( Ax \leq b \) of \( P_\Phi \) with rational data \( A, b \) and initially, that either \( P_\Phi = \emptyset \) or \( \text{dim} \ P_\Phi = n \). The case of flat polyhedra is discussed separately. It follows from point 7.5(d) that the ball \( B(0, R) \) contains \( P_\Phi \), where \( R = \sqrt{n}2^\phi \) and
\[
\Phi = (c) + 8n\phi + 2n^2\phi + 2.
\]

The center of \( B(0, R) \) is \( x^0 = 0 \). Checking \( x^0 \in P_\Phi \) we either find an inequality \( a^0x \leq a_0 \) of the linear description of \( P_\Phi \) such that \( a^0x^0 > a_0 \) or \( x^0 \in P_\Phi \) and we are done. If \( a^0x^0 > a_0 \) then \( P_\Phi \subseteq B(0, R) \cap \{ x \in \mathbb{R}^n : a^0x \leq a_0 \} \). Replacing \( a_0 \) by \( a^0x^0 \) we have that
\[
P_\Phi \subseteq S_1 = B(0, R) \cap \{ x \in \mathbb{R}^n : a^0x \leq a^0x^0 \} \subseteq E_1,
\]
where \( E_1 \) is an ellipsoid (of minimum volume) that contains \( S_1 \). Let \( x^1 \), the center of \( E_1 \), be the next “trial” solution: if \( x^1 \in P_\Phi \) we are done. Otherwise, we find an inequality \( a^1x \leq a_1 \) from the linear description of \( P_\Phi \) such that \( a^1x^1 > a_1 \) and iterate. At the \( k^th \) iteration of this algorithm we have the center \( x^k \) of an ellipsoid \( E_k = E_{Q_k}(x^k, 1) \), where \( Q_k = F_kF_k^T \) is a positive definite matrix defining \( E_k \). By construction \( P_\Phi \subseteq E_k \). Either \( x^k \in P_\Phi \) -- in which case we are done -- or we find an inequality \( a^Tx \leq a_0 \) belonging to the linear description of \( P \) or \( P_\Phi \) such that \( a^Tx^k > a_0 \). In this case we set
\[
x^{k+1} = x^k - \frac{1}{n+1}F_kd \quad \text{where} \quad d = \frac{F_k^Ta}{\|F_k^Ta\|}, \tag{9.1}
\]
\[
F_{k+1} = \sqrt{\frac{n^2}{n^2-1}}F_k \left( I_n - \left(1 - \frac{n-1}{n+1}\right)dd^T\right). \tag{9.2}
\]

We get an ellipsoid \( E_{k+1} = E_{Q_{k+1}}(x^{k+1}, 1) \) with center \( x^{k+1} \) and positive definite matrix \( Q_{k+1} = F_{k+1}F_{k+1}^T \) defining \( E_{k+1} \). As shown in Chapter 9.2, \( E_{k+1} \supseteq P_\Phi \) and
\[
\text{vol}(E_{k+1}) \leq e^{-1/2n}\text{vol}(E_k). \tag{9.3}
\]

---

1 Divide and conquer!
Iterating \( k \) times, we get \( \text{vol}(E_k) \leq V_0 e^{-k/2n} \) for \( k \geq 0 \), where \( V_0 \) is the volume of \( B(0,R) \). Unless the algorithm stops with \( x^k \in P_\Phi \) for some \( k \), it suffices to iterate at most

\[
k_E = \lceil 2n(\log V_0 - \log V_{P_\Phi}) \rceil
\]
times to conclude that \( P_\Phi = \emptyset \), where \( V_{P_\Phi} \) is the volume of \( P_\Phi \). By assumption \( P \) is either empty or full-dimensional. If \( P \neq \emptyset \), we can bound \( V_{P_\Phi} \) from below because \( P \) is a rational polyhedron.

In the left part of Figure 9.1 we show the iterative application of the ellipsoid algorithm when applied to the data of Exercise 8.2 (ii) without the objective function. We start at the point \( x^0 = (60, 60) \) as the center of the initial ball with radius \( \|x^0\| \) which contains all of the feasible set. We select as the “next” inequality \( a^T x \leq a_0 \) the one for which the slack \( a^T x^0 - a_0 \) is largest, to get \( x^1 \); etc.

The above formulas yield polynomial step complexity, but not polynomial time complexity of the calculation. For the latter it is necessary that the digital sizes of \( n \) and \( \phi \) stay bounded by a polynomial function of \( n \), \( \phi \) and \( e \). It must also be shown that all necessary calculations can be carried out in approximate arithmetic, i.e. on a computer with limited word size. In the left part of Figure 9.1 we pretended that we can compute (9.1) and (9.2) “perfectly” – even though we divide, take square roots and calculate on a computer with a word size of merely 64 bits. To be correct, we have to replace the equality signs in (9.1) and (9.2) by the \( \approx \) sign and specify the precision with which we need to calculate the corresponding numbers.

The geometric idea for the way to deal with the problem of approximate calculations is shown in the right part of Figure 9.1 for the ellipsoid \( E_1 \). Since we cannot compute the center \( x^1 \) of \( E_1 \) by formula (9.1) exactly, we get an approximate center \( x^A \) by committing round-off errors. To approximate the matrix \( F_1 \) given by (9.2) we multiply the right-hand side by some factor \( \beta \geq 1 \), i.e. we scale all elements of it up to make them bigger. An approximate computation with round-off errors yields a matrix \( F_A \) that is used in lieu of \( F_1 \). This corresponds to “blowing up” the perfect arithmetic ellipsoid \( E_1 \) concentrically to the ellipsoid \( E_\beta \) of Figure 9.1, i.e. \( E_\beta \) is a homothetic image of \( E_1 \) with a factor \( \beta \geq 1 \) of dilatation. The approximate calculation of \( F_A \) is then carried out with a sufficient precision to guarantee that the ellipsoid \( E_A \) with center \( x^A \) and defining matrix \( Q_A = F_A F_A^T \) contains the ellipsoid \( E_1 \) completely. In Chapter 9.2 we show that a blow-up factor of

\[
\beta = 1 + 1/12n^2
\]

works, where \( n \) is the number of variables of the optimization problem. Our graphical illustration in Figure 9.1 is “artistic”. We used \( \beta \approx \sqrt{1.5} \) to produce the figure and not \( \beta = 49/48 \), which is a lot smaller and works in \( \mathbb{R}^2 \).

In every iteration the ellipsoid algorithm needs only one inequality that is violated or the message that a violated inequality does not exist, i.e., at every iteration we have to solve a separation problem (or constraint identification problem) for the polyhedron \( P \) like the one we discussed in point 7.5(h). So far we have assumed that the number of constraints of the linear description of \( P \) does not matter and that the problem of finding a violated inequality can e.g. be done by listing and checking every single one of them. We call this method LIST-and-CHECK. LIST-and-CHECK does not give a polynomial algorithm for the linear optimization over rational polyhedra if the number of constraints of the linear description of \( P \) is exponential in \( n \). But assuming the existence of some polynomial-time algorithm for the separation problem we get the existence of a polynomial-time algorithm for the optimization problem over rational polyhedra – namely the ellipsoid algorithm. It is a nontrivial result that the reverse statement holds as well:
if for some rational polyhedron $P$ the optimization problem can be solved in polynomial time, then the separation problem for $P$ can be solved in polynomial time as well. This equivalence of optimization and separation for rational polyhedra is of fundamental importance in itself and particularly important for the field of combinatorial optimization: it constitutes the theoretical backbone of the algorithmic approach to combinatorial optimization problems called branch-and-cut.

### 9.1 Matrix Norms, Approximate Inverses, Matrix Inequalities

We have to “truncate” numbers in the ellipsoid algorithm and thus to replace e.g., a matrix $F$ by some matrix $F_A$, say, satisfying $F_A \approx F$. To analyze such an approximation we need a norm. The **Frobenius norm** of an $m \times n$ matrix $F$ of real numbers $f_{ij}$ is

$$
\|F\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij})^2}.
$$

(9.4)

Other norms that are frequently encountered in numerical linear algebra are

$$
\|F\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |f_{ij}|, \quad \|F\|_2 = \max \{|\langle Fx, y \rangle \} : \|x\| = 1\}, \quad \|F\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |f_{ij}|.
$$

$\|F\|_2$ is the **spectral norm** and its value equals the square root of the largest eigenvalue of $F^T F$. If $F$ is nonsingular, then $\|F^{-1}\|_2 = \lambda^{-1}(F)$, i.e., it is the reciprocal of the square root of the smallest eigenvalue of $F^T F$. There are a number of relationships between these various matrix norms, e.g.,

$$
\|F\|_2 \leq \|F\|_F \leq \sqrt{n} \|F\|_2.
$$

The spectral norm yields probably the most elegant proofs for what is to follow, but it is also the hardest one to compute. To avoid issues of computation we will not use it. We use the Frobenius norm and simply drop the subscript $F$ for notational convenience. In Exercise 9.2 most of the...
properties of the Frobenius norm that we need are stated. In particular, for any two \(m \times n\) real matrices we have
\[
\|FR\| \leq \|F\|\|R\|, \quad \|F(I_n - \alpha rr^T)\|^2 = \|F\|^2 - \alpha(2 - \alpha\|r\|^2)\|Fr\|^2.
\] (9.5)

**Remark 9.1** Let \(R\) be any \(n \times n\) matrix of reals with \(\|R\| < 1\). Then \((I_n - R)^{-1}\) exists and
\[
\|(I_n - R)^{-1}\| \leq \frac{1}{1 - \|R\|}.
\] (9.6)

Let the elements of \(F_A\) be obtained by truncating the elements of some nonsingular matrix \(F\), i.e., \(F_A \approx F\), and \(F_A = F + R\). Thus \(R\) is the matrix of “errors” due to the rounding or the truncation and the Frobenius norm is the sum of the squared errors for each element of \(F_A\). We need to know when \(F_A\) is nonsingular and how the errors “propagate” into the inverse of \(F_A\).

**Remark 9.2** Let \(F\) be any nonsingular matrix of size \(n \times n\) and \(R\) be any \(n \times n\) matrix with \(\|F^{-1}R\| < 1\). Then \((F + R)^{-1}\) exists and satisfies the inequalities
\[
\|(F + R)^{-1}\| \leq \frac{\|F^{-1}\|}{1 - \|F^{-1}R\|},
\] (9.7)
\[
\|(F + R)^{-1} - F^{-1}\| \leq \frac{\|R\|\|F^{-1}\|^2}{1 - \|F^{-1}R\|}.
\] (9.8)

To carry out the analysis of the ellipsoid algorithm using approximate arithmetic, we need the following two inequalities for the determinant and the norm of the inverse of a nonsingular matrix repeatedly.

**Remark 9.3** (i) Let \(F\) be any \(n \times n\) matrix of reals. Then we have the inequality
\[
|\text{det}F| \leq n^{-n/2}\|F\|^n.
\] (9.9)

(ii) If \(F\) is nonsingular and \(n \geq 2\), then we have the inequality
\[
\|F^{-1}\| \leq n(n - 1)^{-\frac{n-1}{2}}\|F\|^{n-1}/|\text{det}F|.
\] (9.10)

### 9.2 Ellipsoid “Halving” in Approximate Arithmetic

To carry out the various constructions analytically, we drop the index \(k\) and denote by \(F\) the nonsingular matrix defining the “current” ellipsoid \(E_0\) and by \(x^0\) its center, i.e.,
\[
E_0(x^0, 1) = \{x \in \mathbb{R}^n : \|F^{-1}(x - x^0)\| \leq 1\}.
\] (9.11)

Let \(a^Tx \leq a_0\) be the linear inequality that the algorithm identifies and denote by \(x^P\) and \(F_P\) the updates (9.1) and (9.2) that result if calculated in perfect arithmetic, i.e. with an infinite precision:
\[
x^P = x^0 - \frac{1}{n + 1}Fd \quad \text{where} \quad d = \frac{F^Ta}{\|F^Ta\|}.
\] (9.12)
9.2. ELLIPSOID “HALVING” IN APPROXIMATE ARITHMETIC

\[ F_P = \sqrt{\frac{n^2}{n^2 - 1}} F \left( I_n - \left(1 - \sqrt{\frac{n - 1}{n + 1}}\right) d d^T \right). \]  

(9.13)

Assuming that \( F_P \) is nonsingular, we get in perfect arithmetic an ellipsoid \( E_P \) with center \( x^P \) which in the iterative scheme of the introduction is the “next” ellipsoid, i.e.

\[ E_P(x^P, 1) = \{ x \in \mathbb{R}^n : \| F_P^{-1}(x - x^P)\| \leq 1 \}. \]  

(9.14)

For any “blow-up” factor \( \beta \geq 1 \) denote by \( E_\beta \) the enlarged ellipsoid with center \( x^P \), i.e.,

\[ E_\beta = E_P(x^P, \beta) = \{ x \in \mathbb{R}^n : \| F_P^{-1}(x - x^P)\| \leq \beta \}. \]  

(9.15)

Thus the enlarged ellipsoid \( E_\beta \) is defined with respect to the matrix

\[ F_\beta = \beta F_P. \]  

(9.16)

Because of the finite wordlength of the computer we commit round-off errors and compute approximately

\[ x^A \approx x^P, \quad F_A \approx F_\beta. \]  

(9.17)

Assume that the error in the approximate calculation satisfies

\[ \| x^A - x^P \| \leq \delta \quad \text{and} \quad \| F_A - F_\beta \| \leq \delta \quad \text{where} \quad \delta \leq p(n) \frac{|\text{det}F|}{\|F\|^{n-1}}, \quad p(n) = 10^{-4n^2}. \]  

(9.18)

From inequality (9.9) it follows that the error of the approximation is less than \( \|F\| \) because by (9.9) e.g. \( \| x^A - x^P \| \leq p(n)n^{-n/2}\|F\| \). Condition (9.18) is translated in Chapter 9.3 into the number of digits of each component of \( x^A \) and \( F_A \) that need to be calculated correctly before and after the “decimal” point in binary arithmetic. The calculations (9.17) yield an approximation \( E_A \)

\[ E_A(x^A, 1) = \{ x \in \mathbb{R}^n : \| F_A^{-1}(x - x^A)\| \leq 1 \}, \]  

(9.19)

to \( E_\beta \) which is an ellipsoid if \( F_A \) is “close enough” to \( F_\beta \) so as to guarantee the nonsingularity of \( F_A \).

The “battle-plan” of the proof is as follows:

- Establish that \( F_P \) is nonsingular if \( F \) is nonsingular and that (9.3) remains correct.
- Show that if \( E_0 \) contains \( P \) or \( P_\Phi \) then so does the ellipsoid \( E_P \).
- Show that if (9.18) is satisfied neither \( x^A \) nor \( F_A \) “grow” too much in size.
- Assuming (9.18) show that \( F_A \) is nonsingular and that \( E_A \supseteq E_P \).
- Assuming (9.18) show that \( \frac{\text{vol}(E_A)}{\text{vol}(E_\beta)} \) satisfies a relation like (9.3) to conclude a polynomial running time of the approximate calculations.
- Establish a lower bound on the volume \( V_\Phi \) if \( \text{dim}P_\Phi = n \) and prove inductively that (9.18) is satisfied for all necessary iterations. This is done in the next section.
The first step is easy because

\[
\det F_P = \left( 1 + \frac{1}{n} \right)^{- \frac{n+1}{2}} \left( 1 - \frac{1}{n} \right)^{- \frac{n-1}{2}} \det F .
\]  

(9.20)

The factor appearing in (9.20) satisfies

\[
\left( 1 + \frac{1}{n} \right)^{- \frac{n+1}{2}} \left( 1 - \frac{1}{n} \right)^{- \frac{n-1}{2}} \leq e^{- \frac{n}{4}} \text{ for all } n \geq 1.
\]  

(9.21)

Computing the volumes of \( E_0 \) and \( E_P \) we get from (7.23) and (9.20)

\[
\frac{\text{vol}(E_P)}{\text{vol}(E_0)} = \left( 1 + \frac{1}{n} \right)^{- \frac{n+1}{2}} \left( 1 - \frac{1}{n} \right)^{- \frac{n-1}{2}} \leq e^{- \frac{n}{4}}
\]  

by (9.21) and (9.3) follows. By (9.20) \( F_P \) is nonsingular and its inverse in terms of \( F^{-1} \) is

\[
F_P^{-1} = \sqrt{\frac{n^2 - 1}{n^2} \left( I_n - \left( 1 - \sqrt{\frac{n+1}{n-1}} \right) \beta d^T \right)} \det F^{-1} .
\]  

(9.22)

For notational convenience define \( Q = FF^T \) and \( Q_P = F_P F_P^T \), i.e. \( Q \) and \( Q_P \) are the positive definite matrices defining \( E_0 \) and \( E_P \).

\[
Q_P = \frac{n^2}{n^2 - 1} Q \left( I_n - \frac{2}{n+1} \frac{aa^T Q}{a^T Q a} \right), \quad Q_P^{-1} = \frac{n^2 - 1}{n^2} \left( Q^{-1} + \frac{2}{n-1} \frac{aa^T}{a^T Q a} \right) .
\]  

(9.23)

**Remark 9.4**

(i) For all \( x \in E_0 \) and \( a \in \mathbb{R}^n, a \neq 0 \), \( -\sqrt{a^T Q a} \leq a^T (x - x^0) \leq \sqrt{a^T Q a} \).

(ii) Let \( a^T x \leq a_0 \) with \( a \neq 0 \) be any inequality such that \( a^T x^0 \geq a_0 \) and \( X \subseteq E_0 \) be any subset of \( E_0 \) such that \( a^T x \leq a_0 \) for all \( x \in X \). Then \( X \subseteq E_P \).

By inductive reasoning it follows that \( P_\beta \subseteq E_k \) if \( B(0, R) \) has a large enough radius to contain \( P_\beta \) initially.

To justify the approximate calculations of \( x^A \) and \( F_A \), we compute like in Exercise 9.2 (iii) from (9.16), (9.13) and (9.22)

\[
\| F_\beta \| \leq \beta \sqrt{\frac{n^2}{n^2 - 1}} \| F \|, \quad \| F_P^{-1} \| \leq \beta^{-1} \frac{n+1}{n} \| F^{-1} \| ,
\]  

(9.24)

where we have used \( \| d \| = 1 \), see (9.12). In the following we use the inequality

\[
2^x \geq 1 + \frac{2}{3} x + \frac{2}{9} x^2 \quad \text{for all } x \geq 0 .
\]  

(9.25)

**Remark 9.5** If (9.18) is satisfied, then for \( \beta = 1 + 1/12n^2 \) and all \( n \geq 2 \)

\[
\| x^A \| \leq \| x^0 \| + \frac{1}{n} \| F \|, \quad \| F_A \| \leq 2^{1/n^2} \| F \|. \]  

(9.26)
The following inequality is readily verified for all \( n \geq 2 \)
\[
1 + 2(n + 1)(n - 1)^{-\frac{1}{2}} n^{-2} 10^{-4} \leq 1 + \frac{1}{12n^2} .
\] (9.27)

**Remark 9.6** If (9.18) is satisfied and \( \det F \neq 0 \), then for \( \beta = 1 + 1/12n^2 \) and all \( n \geq 2 \) the matrix \( F_A \) is nonsingular and \( E_A(x^A, 1) \geq E_P(x^P, 1) \).

The main point in the proof that \( E_P(x^P, 1) \subseteq E_A(x^A, 1) \) is the estimation
\[
\| F_A^{-1}(x - x^A) \| \leq (1 + \| F_A^{-1} \| \| F_\beta - F_A \|) (\| F_\beta^{-1}(x - x^P) \| + \| F_\beta^{-1}(x^P - x^A) \|).
\] (9.28)

By Remark 9.4 the feasible set is contained in the approximate ellipsoid if (9.18) is satisfied. The verification of (9.18) will be done by induction.

To estimate next \( \frac{\text{vol}(E_A)}{\text{vol}(E_0)} \) we calculate using (9.4), (9.13), (9.16) and (9.22)
\[
|| F^{-1} F_\beta || = \beta \sqrt{n \left( 1 - \frac{1}{(n + 1)^2} \right)} , \quad || F^{-1} F ||_\infty = \frac{\sqrt{n^3 + n + 2}}{\beta n} .
\] (9.29)

**Remark 9.7** If \( \det F \neq 0 \) and (9.18) is satisfied, then for \( \beta = 1 + 1/12n^2 \) and all \( n \geq 2 \)
\[
2^{-1/n} |\det F| \leq |\det F_A| \leq 2^{-1/4n} |\det F| .
\] (9.30)

If (9.18) is satisfied at every iteration we can calculate the number of iterations required by the ellipsoid algorithm using approximate arithmetic, i.e. when (9.1) and (9.2) are replaced by (9.17).

From (7.23) we compute
\[
\frac{\text{vol}(E_A)}{\text{vol}(E_0)} = \frac{|\det F_A|}{|\det F|} \leq 2^{-1/4n} < e^{-1/6n}
\]

using Remark 9.7 and (9.25). Denoting the ellipsoid at the \( k^{th} \) iteration again by \( E_k \), we get \( \text{vol}(E_k) < V_0 e^{-k/6n} \) and consequently, after at most
\[
k_A = \lceil 6n(\log V_0 - \log V_{P_k}) \rceil
\]
iterations the ellipsoid algorithm with approximate arithmetic stops with the message that \( P_\Phi = \emptyset \) - unless for some \( k < k_A \) the corresponding iterate \( x^k \) belongs to \( P_\Phi \). Here we have assumed that (9.18) is satisfied at every iteration and that \( P_\Phi \neq 0 \) implies \( \dim P_\Phi = n \) and that \( \text{vol}(P_\Phi) \geq V_{P_k} \) - assumptions that we will have to remove later.

### 9.3 Polynomial-Time Algorithms for Linear Programming

Having settled the analytical problem of calculating the “perfect” arithmetic ellipsoid approximately by permitting round-off errors let us consider the linear program
\[(LP) \quad \max \{ cx : Ax \leq b \} ,\]
where \( c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \) all have integer components only, \( m \geq 1, n \geq 2 \) and \( A \neq 0 \) to rule out trivialities. The problem (LP) has only inequalities, i.e. any equations have been eliminated or replaced by their corresponding pairs of inequalities.
From Chapter 7.5.1 we know that the integrality assumption is polynomially equivalent to assuming that the data are rational. So the integrality assumption is convenient, but not necessary. Other than this one we make no assumption. The rank of $A$ can be anywhere between 1 and $\min\{m,n\}$ and the feasible set

$$\mathcal{X} = \{x \in \mathbb{R}^n : Ax \leq b\} \quad (9.31)$$

can be a solid, flat, pointed or blunt polyhedron of $\mathbb{R}^n$. Nonnegativity requirements are part of $(A,b)$.

To demonstrate the polynomial solvability of (LP) – polynomial in terms of $m$, $n$, the digital size $\langle c \rangle$ of the vector $c$ and the facet complexity $\phi$ of $\mathcal{X}$ – we proceed in three steps:

- We assume that $\mathcal{X}$ is either empty or bounded and full dimensional. By running the “basic” ellipsoid algorithm we decide whether or not $\mathcal{X} = \emptyset$ by producing a rational vector $x \in \mathcal{X}$ if $\mathcal{X} \neq \emptyset$.
- We remove the assumptions and show that by embedding any $\mathcal{X}$ into a somewhat larger polyhedron $\mathcal{X}_h$ we can answer the question in polynomial time for any rational $\mathcal{X}$.
- We use binary search like in Chapter 7.5.3 to prove the existence of a polynomial-time algorithm for the linear program (LP).

To carry out the first step we make the following assumption.

**Assumption A:** There exists $R \geq 1$, i.e. some radius, such that $\mathcal{X} \subseteq B(0,R)$. Moreover, if $\mathcal{X} \neq \emptyset$ then there exists $x \in \text{relint}\mathcal{X}$ and a radius $r > 0$ such that $B(x,r) \subseteq \mathcal{X}$.

To state the basic ellipsoid algorithm, we denote by $x^P$, $x^k$ the components of $x^P$, $x^k$ and by $\beta f_j^l$, $k f_j^l$ the elements of $F_{\beta}$, $F_k$ for $1 \leq i, j \leq n$ and $k \geq 0$. The input consists of $m$, $n$, $A$ and $b$, the parameters $R$, $T$ and $p$ and if $\mathcal{X} \neq \emptyset$, $x$ is the output of the algorithm.

**Basic Ellipsoid Algorithm** $(m, n, R, T, p, A, b, x)$

**Step 0:** Set $x^0 := 0$, $F_0 := R I_n$, $k := 0$.

**Step 1:** If $k \geq T$, stop “$\mathcal{X}$ is empty”.

- If $a^i x^k \leq b_i$ for all $1 \leq i \leq m$, stop “$x := x^k$”.
- Let $(a^i, b_i)$ for some $i \in \{1, \ldots, m\}$ be such that $a^i x^k > b_i$ and set $a^T := a^i$.

**Step 2:** Calculate approximately $x^{k+1} \approx x^P$ and $F_{k+1} \approx F_{\beta}$ where

$$x^P := x^k - \frac{1}{n+1} \frac{F_k F_k^T a}{\|F_k^T a\|} , \quad (9.32)$$

$$F_{\beta} := \frac{n+1/2n^{1/2}}{\sqrt{n}-1} F_k \left( I_n - \frac{1 - \sqrt{(n-1)/(n+1)}}{a^T F_k F_k^T a} (F_k^T a)(a^T F_k) \right) , \quad (9.33)$$

such that the binary representation of each component of $x^{k+1}$ and $F_{k+1}$ satisfies $|x^{k+1}_j - x^P_j| \leq 2^{-p}$ and $|k+1 f_j^l - \beta f_j^l| \leq 2^{-p}$ for $1 \leq i, j \leq n$. Replace $k + 1$ by $k$ and go to Step 1.

**Remark 9.8** (Correctness and finiteness) If Assumption A is true, then the basic ellipsoid algorithm finds a rational vector $x \in \mathcal{X}$ or concludes correctly that $\mathcal{X} = \emptyset$ when it is executed with the parameters

$$T = \lceil 6n^2 \log \frac{R}{r} \rceil , \quad p = 14 + n^2 + \lceil 15n \log \frac{R}{r} \rceil . \quad (9.34)$$
9.3. POLYNOMIAL-TIME ALGORITHMS FOR LINEAR PROGRAMMING

The proof uses Remarks 9.5, 9.6 and 9.7 to estimate \( \text{inter alia} \)
\[
\|F_k\| \leq \sqrt{nR^2k}, \quad \|x^k\| \leq kR^2, \quad R^2 \leq |\det F_k| \leq R^2. \tag{9.35}
\]
We are now ready to drop Assumption A. Denote by \( \phi \) the facet complexity of \( X \), i.e.
\[
\phi \geq \max_{1 \leq i \leq m} \{ \langle a_i \rangle \} \quad \text{and} \quad \phi_A = \max_{1 \leq i \leq m} \langle a_i \rangle. \tag{9.36}
\]
It follows that \( n + 1 \leq \phi_A < \phi \) since \( A \neq 0 \) and moreover, we can choose \( \phi \) such that \( \phi \leq \phi_A + \langle b \rangle \), where \( \langle b \rangle \) is the digital size of \( b \). For any integer \( h \geq 1 \) we denote by \( h \in \mathbb{R}^m \) the vector having \( m \) components equal to \( 1/h \) and let
\[
X_h = \{ x \in \mathbb{R}^n : Ax \leq b + h^{-1} \}, \tag{9.37}
\]
which corresponds to “perturbing” the feasible set \( X \) of (LP).

**Remark 9.9**

(i) \( X \neq \emptyset \) if and only if \( X_h \neq \emptyset \) for all \( h \geq p2^{n \phi_A} \) where \( p = 1 + \min\{m, n\} \).

(ii) If \( X \neq \emptyset \), then for all \( u \geq 2^{n \phi} \) and all integers \( h \geq 1 \) the set \( X_h^u \) of solutions to
\[
Ax \leq b + h^{-1}, \quad -u - 1/h \leq x_j \leq u + 1/h, \quad \text{for} \ 1 \leq j \leq n \tag{9.38}
\]
is bounded, full dimensional and \( B(x, r_h) \subseteq X_h^u \) for all \( x \in X \) where
\[
r_h = h^{-1}2^{-n \phi + n}. \tag{9.39}
\]
To visualize the construction of Remark 9.9 take for instance
\[
X = \{ x \in \mathbb{R}^2 : 1 \leq x_1 \leq 1, 1 \leq x_2 \leq 1 \}
\]
and bring it into the form (9.37). Graphing the corresponding solution set, one sees that the introduction of the perturbation \( 1/h \) in each inequality corresponds to a “tearing apart” of the solution set to obtain a full dimensional set of solutions; see also Figure 9.8 below. If \( X \) is empty then there is nothing to tear apart and as the first part of Remark 9.9 shows, the emptiness of \( X_h \) is preserved if the perturbation is “small enough”. Running the basic ellipsoid algorithm with \( R = 2^{n \phi}, T = 20n^3 \phi \) and \( p = 55n^2 \phi \) we conclude like in point 7.6(g):

**Remark 9.10**

Every \( m \times n \) system of linear inequalities \( Ax \leq b \) with rational data can be “solved” in time that is polynomial in the digital size of its input.

There are several ways to deal with the optimization aspect of the linear program (LP). The simplest way is to use linear programming duality and to reduce the optimization problem to the problem of finding a solution to a system of linear inequalities – like we did in Remark 6.5. The second way uses binary search and a “sliding objective” function; see the text.

**Remark 9.11**

Every linear program with rational data can be “optimized” in time that is polynomial in the digital size of its input.

Neither the radius \( R \) of the ball circumscribing \( X \) or \( X_h^u \) nor the radius \( r \) of the ball that is inscribed into \( X \) or \( X_h^u \), see (9.39), depend on the number \( m \) of linear inequalities of (LP). Consequently, none of the other two parameters \( T \) and \( p \) of the basic ellipsoid algorithm depends
on the number \( m \). They are polynomial functions of \( \phi, n \) and \( \langle c \rangle \) only. The dependence of the basic ellipsoid algorithm on the number \( m \) of the inequalities of (LP) enters in Step 1 when we have to find a violated inequality for the system (9.43) or prove that none exists. The same is true for the auxiliary computations. For the time being we assume that we find violated inequalities by the “algorithm” LIST-and-CHECK that we discussed in the introduction. If \( m \) is of the order of \( n \), i.e. \( m = O(n) \), then the total effort to solve (LP) becomes a polynomial function of \( n, \phi \) and \( \langle c \rangle \) only, whereas in the general case we need, of course, note the dependence on \( m \) explicitly.

Before coming back to the question of how to deal with the case of possibly exponentially many constraints defining \( \mathcal{X} \) we first discuss some “practical” variants of the basic ellipsoid algorithm.

### 9.4 Deep Cuts, Sliding Objective, Large Steps, Line Search

Going back to Figure 9.1 we see that instead of cutting \( E_k \) with \( a^T x \leq a_0 \) where the right-hand side equals \( a_0 \) and which is valid for all \( x \in \mathcal{X} \) – we replaced \( a_0 \) by the larger quantity \( a^T x_k \). This replacement forces the cut to pass through the center of the current ellipsoid and the resulting algorithm is therefore called central cut ellipsoid algorithm. It is not overly difficult to work out the formulas corresponding to (9.1) and (9.2) when instead of \( a^T x \leq a^T x_k \) we use the deep cut \( a^T x \leq a_0 \). They are given below.

A less obvious modification of the basic algorithmic idea concerns the optimization aspect of (LP). Let

\[
z = \max \{ cx_k : x_k \text{ feasible} \},
\]

where initially \( z = -\infty \). Then we can use the objective function as a “sliding constraint” of the form \( cx \geq z \) where the value of \( z \) increases during the course of the calculations. This gives rise to a sliding objective and thereby to a device that speeds the convergence of the procedure considerably.

A third modification to the basic idea goes as follows. Suppose the current iterate \( x_k \) is feasible. Then the point \( x^* = x^k + \frac{F_k F_k^T c^T}{\|cF_k\|} \) maximizes the linear function \( cx \) over the current ellipsoid \( E_k = \{ x \in \mathbb{R}^n : \| F_k^{-1} (x - x_k) \| \leq 1 \} \); see Remark 9.4(i). Consequently, we can determine by a least ratio test the largest \( \lambda \geq 0 \) such that

\[
x(\lambda) = x^k + \lambda (x^* - x^k)
\]

and thereby make a large step towards optimality by “shooting” through the interior of the feasible set. We calculate the largest \( \lambda \), the corresponding feasible \( x(\lambda) \) and its objective function value \( z_\lambda \), say. If \( z_\lambda > z \), then we update the current best solution to be \( x(\lambda) \), replace \( z \) by \( z_\lambda \) and use in one of the subsequent iterations the objective function as a cut to reduce the volume of the ellipsoid.

The fourth modification of the basic algorithmic idea is aimed at improving the chances of the algorithm to find feasible solutions to (LP). The algorithm generates a sequence of individual points and their probability to fall into the feasible set is rather small. Consider two consecutive centers \( x^k \) and \( x^{k+1} \) of the ellipsoids generated by the algorithm. They determine a line

\[
x(\mu) = (1 - \mu)x^k + \mu x^{k+1} \text{ where } -\infty < \mu < +\infty.
\]

We can decide the question of whether or not \( x(\mu) \) meets the feasible set by a line search that involves again a simple least ratio test. If the test is negative, we continue as we would do without
it. If the test comes out positive, then we get an interval $[\mu_{\text{min}}, \mu_{\text{max}}]$ such that $x(\mu)$ is feasible for all $\mu$ in the interval. Computing the objective function we find $cx(\mu) = cx^k + \mu(cx^{k+1} - cx^k)$. Consequently, if $cx^{k+1} > cx^k$ then $\mu = \mu_{\text{max}}$ yields the best possible solution vector while $\mu = \mu_{\text{min}}$ does so in the opposite case. The rest is clear: we proceed like we did in the case of large steps.

We are now ready to state an ellipsoid algorithm for linear programs in canonical form

\[
(LP_C) \quad \max \{ cx : \tilde{A} x \leq \tilde{b}, \ x \geq 0 \},
\]

where $(\tilde{A}, \tilde{b})$ is an $m \times (n+1)$ matrix of rationals. We assume that $\tilde{A}$ contains no zero row and denote by $(a^i, b_i)$ for $1 \leq i \leq m+n$ the rows of the matrix $(A, b) = \begin{pmatrix} \tilde{A} & \tilde{b} \\ -I_n & 0 \end{pmatrix}$.

The DCS ellipsoid algorithm takes $m, n, A, b, c$ as inputs. $z_L$ is a lower bound, $z_U$ an upper bound on the optimal objective function value. $R$ is a common upper bound on the variables of $(LP_C)$ and $\varepsilon$ a perturbation parameter to ensure full dimensionality of the feasible set when intersected with the sliding objective function constraint $cx \geq z$. Since we are perturbing the constraint set of $(LP_C)$ by a parameter $\varepsilon > 0$ we shall call solutions to the perturbed constraint set nearly feasible or $\varepsilon$-feasible solutions and correspondingly, we shall utilize the term $\varepsilon$-optimal solution to denote a nearly feasible, nearly optimal solution to $(LP_C)$. $V_F$ is a positive lower bound on the volume of a full dimensional, $\varepsilon$-optimal set. In other words, if the current ellipsoid has a volume less than $V_F$ we shall conclude that either $\varepsilon$-optimality is attained – if a feasible solution $x$ with objective function value $z$ was obtained – or else that the feasible set of $(LP_C)$ is empty. As we know from Chapters 7.5 and 9.3 we can almost always find theoretical values for $\varepsilon$ and $R$ and by consequence for $z_L$, $z_U$ and $V_F$ as well that the algorithm needs to converge.

In practice, we set the perturbation parameter e.g. $\varepsilon = 10^{-4}$ and use a rough data dependent estimate for the common upper bound $R$ on the variables. Similarly, we use e.g. $V_F = 10^{-2}$ to fix the stopping criterion and from $R$ we estimate $z_L$ and $z_U$ e.g. as follows

\[
z_L = -1 + nc^- R, \quad z_U = 1 + nc^+ R,
\]

where $c^- = \min\{c_j : 1 \leq j \leq n\}$ and $c^+ = \max\{c_j : 1 \leq j \leq n\}$.

“DCS” stands for deep cut, sliding objective, large steps and line search, i.e. all of the devices that we discussed above to speed the empirical rate of convergence of the underlying basic algorithmic idea. For this “practical” version of the ellipsoid algorithm we ignore the blow-up factor $\beta \geq 1$ that is necessary to obtain the theoretical result since $\beta - 1 = 1/12n^2$ is a horribly small positive number for reasonably sized $n$.

In the DCS ellipsoid algorithm we assume $c \neq 0$. If $c = 0$ then some modifications and simplifications impose themselves the details of which we leave as an exercise for you to figure out.

**DCS Ellipsoid Algorithm** $(m, n, z_L, z_U, \varepsilon, R, V_F, A, b, c, x, z)$

**Step 0:** Set $k := 0$, $x_0 := R/2$ for $1 \leq j \leq n$, $z := z_L$, $z_0 := z_L$.

\[
R_0 := \sqrt{n}(1 + R/2), \quad H_0 := R_0 I_n, \quad f_0 := \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} R_0 = R_0^{\frac{n}{2}} / \Gamma(1 + n/2).
\]

**Step 1:** Set $mxv := b_j + \varepsilon - a^i x^k$ where $b_j - a^i x^k \leq b_i - a^i x^k$ for all $1 \leq i \leq n + m$.

- **if** $mxv < 0$ **go to** Step 2.
- **Set** $x := x^k + H_i H_i^T c^T / \|cH_i\|$, $\lambda := \max\{\lambda : \lambda a^i (x^* - x^k) \leq b_i + \varepsilon - a^i x^k, \ 1 \leq i \leq n + m\}$.
- **if** $\lambda \geq 1$ **stop** "$(LP_C)$ is unbounded."
if \( c(x^k + \lambda(x^* - x^k)) \leq z \) go to Step 2.

Set \( \Xi := x^k + \lambda(x^* - x^k) \), \( z := c\Xi \).

**Step 2:** if \( (c\Xi > z \text{ or } (\text{max} < 0 \text{ and } z_0 - z > \text{max})) \) then

Choose \( \Theta \) so that \( b_j + \varepsilon \leq \Theta \leq a_i x^k \). Set \( \alpha_k := \frac{a_i^T x^k - \Theta}{\|a_i H_k\|}, r_T := -c, z_0 := z \).

else

Set \( \alpha_k := (z - c\Xi)/\|cH_k\|, r_T := -c, z_0 := z \).

endif

**Step 3:** if \( \alpha_k < 1 \) and \( V_L \geq V_F \) go to Step 4.

if \( z_L < z < z_U \) stop \( \Xi \) is an \( \varepsilon \)-optimal solution to \( (LP_C) \).

stop \( "(LP_C) \) is infeasible or unbounded."

**Step 4:** Set

\[
\begin{align*}
x^{k+1} := x^k &- \frac{1 + n\alpha_k}{(n + 1)\|H_k^T r\|} H_k H_k^T r, \\
H_{k+1} := n &\left[ 1 - \frac{\alpha_k^2}{n^2 - 1} H_k \left( I_n - \left( 1 - \frac{\left( n - 1 \left( 1 + \alpha_k \right) \right) (H_k^T r)(r^T H_k)}{\|H_k^T r\|^2} \right) \right) \right],
\end{align*}
\]

\( V_{k+1} := (1 - \alpha_k^2)^{\frac{n+1}{2}} (1 - \alpha_k) f_0 V_k. \)

Let \( I := \{ \mu \in \mathbb{R} : \mu a_i(x^{k+1} - x^k) \leq b_i + \varepsilon - a_i x^k \text{ for } 1 \leq i \leq n + m \}. \)

if \( I \neq \emptyset \) and \( c\Xi \neq c\Xi + 1 \) then

if \( c\Xi + 1 > c\Xi \) then set \( \Xi := \max\{ \mu : \mu \in I \} \) else \( \Xi := \min\{ \mu : \mu \in I \} \).

if \( \Xi = \infty \) stop \( "(LP_C) \) is unbounded."

if \( \Xi + 1 c\Xi + 1 - c\Xi \) then set \( \Xi := x^k + \Xi(x^{k+1} - x^k), z := c\Xi \).

endif

Replace \( k + 1 \) by \( k \) and go to Step 1.

### 9.4.1 Linear Programming the Ellipsoidal Way: Two Examples

In Figures 9.2, 9.3, 9.4 we show the first nine iterations that result when we use the data of Exercise 8.2 (ii) and minimize \( x_2 \) without the use of the line search, i.e. we assume in Step 4 that always \( I = \emptyset \). In Figures 9.5, 9.6, 9.7 we show the corresponding 12 first iterations when we maximize \( x_2 \) with line search. To make the corresponding pictures more readable we have depicted every third ellipse by a “dashed” curve, whereas all the others are drawn solidly. The first ellipse shown in Figures 9.3, 9.4 and Figures 9.6, 9.7, respectively, are the “last” ellipse of the respective preceding picture. “Dashed” lines correspond to using the original constraints to cut the ellipse in half, while “dotted” lines correspond to cuts using the sliding objective. In Figure 9.3 the arrows show the “large” steps that the algorithm takes, while there are none in Figures 9.5, 9.6, 9.7. Note that in Step 2 we use – like in Step 1 – a “most violated” constraint: a sliding objective cut is executed only if the objective cut is a most violated constraint. Note the different convergence behavior of the algorithm that results from the existence of alternative optima; see the text for a detailed discussion of the computer runs.
Fig. 9.2. Deep cuts, sliding objective, large steps (minimize $x_2$)

Fig. 9.3. Deep cuts, sliding objective, large steps for iterations 3, $\ldots$, 6
**Fig. 9.4.** Proving optimality of a face of dimension 1 in $\mathbb{R}^2$ the ellipsoidal way

**Fig. 9.5.** Deep cuts, sliding objective, line search (maximize $x_2$)

**Fig. 9.6.** Deep cuts, sliding objective, line search for iterations 4, $\ldots$, 8
9.4. DEEP CUTS, SLIDING OBJECTIVE, LARGE STEPS, LINE SEARCH

Fig. 9.7. Proving optimality of a face of dimension 0 in $\mathbb{R}^2$ the ellipsoidal way

9.4.2 Correctness and Finiteness of the DCS Ellipsoid Algorithm

By assumption the parameter $R$ is a common upper bound on the variables of (LP$_C$) that is large enough so that the hypercube

$$\{x \in \mathbb{R}^n : 0 \leq x_j \leq R \text{ for } 1 \leq j \leq n\}$$

contains all of the feasible set of (LP$_C$) if (LP$_C$) is bounded and enough of the unbounded portion of the feasible set to permit us to conclude unboundedness via the value of the objective function; see point 7.5(d) and the discussion of binary search in Chapter 9.3. We start the algorithm at the center $x_0 = \frac{1}{2} Re$ of this hypercube and by choice of $R_0$ in Step 0 the initial ball $B(x_0, R_0)$ does the job.

The validity of the DCS ellipsoid algorithm is established inductively like in Chapter 9.2. Using formula (9.45) for the update $H_{k+1}$ we compute its determinant in terms of the determinant $H_k$

$$\det H_{k+1} = \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n+1}{2}} (1 - \alpha_k^2)^{\frac{n-1}{2}} (1 - \alpha_k) \det H_k . \quad (9.47)$$

To establish the containment of the feasible set in the updated ellipsoid, we form the positive definite matrix $G_{k+1} = H_{k+1} H_{k+1}^T$ and compute its inverse

$$G_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left(G_k^{-1} + \frac{2(1 + n \alpha_k)}{(n - 1)(1 - \alpha_k)} \frac{rr^T}{G_k r} \right) ; \quad (9.48)$$

see Exercise 9.6. Let $E_k = E_k(x^k, 1)$ be the ellipsoid that the DCS algorithm constructs at iteration $k$

$$E_k(x^k, 1) = \{x \in \mathbb{R}^n : (x - x^k)^T G_k^{-1} (x - x^k) \leq 1\} . \quad (9.49)$$

It follows using (7.23) from (9.47) that

$$\frac{\text{vol}(E_{k+1})}{\text{vol}(E_k)} = \left(1 - \alpha_k\right)^{\frac{n+1}{2}} \left(1 + \alpha_k\right)^{\frac{n+1}{2}} \leq e^{-\alpha_k - \frac{1}{n}}$$

for all $k \geq 0$. Setting $V_{k+1} = \text{vol}(E_{k+1})$ and $f_0 = (1 - 1/n)^{-\frac{n+1}{2}} (1 + 1/n)^{-\frac{n+1}{2}}$ we get

$$V_{k+1} = V_0 f_0 \prod_{\ell=0}^k (1 - \alpha_\ell)^{-\frac{n+1}{2}} (1 - \alpha_\ell) , \quad (9.50)$$
which shows that the DCS algorithm updates the volume of the current ellipsoid correctly in formula (9.46). Moreover, it shows the “deflating” effect of the deep cuts on the volume of the ellipsoid quite clearly.

It is shown in the text that the DCS ellipsoid algorithm is correct if the bound $R$ is “large enough” and the perturbation $\varepsilon$ is “small enough”. From (9.50) and the formula for the ratio of the volumina it follows that the stopping criterion $V_k < V_F$ is satisfied after at most

$$[2n \log \frac{V_0}{V_F}]$$

iterations. The $\alpha_i$’s introduce a data-dependency into the stopping criterion that does, however, not change the theoretical worst-case behavior of the algorithm.

9.5 Optimal Separators, Most Violated Separators, Separation

Throughout the rest of this chapter we will deal mostly with rational polytopes $P \subseteq \mathbb{R}^n$ rather than with polyhedra. From Chapter 7.5 we know that we can always do so while preserving polynomiality of the facet complexity and vertex complexity in terms of the original parameters. We are interested in the linear optimization problem $\max \{ cx : x \in P \}$ where the vector $c$ is some row vector with rational coefficients and $P$ has a linear description with possibly exponentially many linear inequalities. To approach this problem we start with a partial linear description of the rational polytope $P$ having $O(n)$ constraints, which gives us a larger polytope $P_0$ that contains $P$. Solving the linear program

$$\max \{ cx : x \in P_0 \}$$

we either conclude that $P_0$ and thus $P$ is empty or we get an optimal solution $x^0 \in P_0$, e.g. an optimal extreme point of $P_0$. Now we check the constraint set of $P$ by some “separation algorithm” - other than LIST-and-CHECK – to find a violated constraint, i.e. we solve something like the separation problem of Chapter 7.5.4 algorithmically. If the separation problem does not produce a violated constraint, then $x^0$ is an optimal solution to the problem $\max \{ cx : x \in P \}$ – see the outer inclusion principle of Chapter 7.5.4 which works with a “local” description of $P$ in the neighborhood of a maximizer $x^{\text{max}}$, say, of $cx$ over $P$ rather than the complete linear description of $P$.

To stress the point that we wish to make once again, suppose that the feasible set of our linear program is given by a convex polygon in $\mathbb{R}^2$ with, say, $10^{10}$ or more “corners” and just as many facets: all you need are at most three of its facet defining constraints to prove optimality of some corner that maximizes your linear function in $\mathbb{R}^2$, see Figure 9.8. The problem is to find the “right” ones and almost nothing else matters.

Suppose that the separation algorithm finds a constraint $h^1 x \leq h^1_0$, say, such that $h^1 x^0 > h^1_0$. Then

$$P \subseteq P_1 = P_0 \cap \{ x \in \mathbb{R}^n : h^1 x \leq h^1_0 \} \subset P_0$$

2“You will depart, you will arrive, you will not die in the war” or is it “you will depart, you will not arrive, you will die in the war”? 
9.5. OPTIMAL SEPARATORS, MOST VIOLATED SEPARATORS, SEPARATION

and we can iterate. The question: does this iterative scheme converge “fast enough” to permit linear optimization over any rational polytope in polynomial time?

This is where the (basic) ellipsoid algorithm enters: neither its running time $T$ nor the required precision $p$, see e.g. (9.34), depend on the number $m$ of the constraints of the linear program. So if we can find a constraint defining $P$ that is violated by the current iterate $x^k$ or prove that no such constraint exists in time that is polynomial in $n$, $\phi$, $(x^k)$ and $(c)$, then the polynomiality of the entire iterative scheme follows from the polynomiality of the ellipsoid algorithm.

Let us discuss first what kind of a violated constraint we wish to find ideally to obtain “fast” convergence of this iterative scheme. We shall forget what other authors have called “the separation problem” and first determine what it is that we really want.

Denote by $P_k$ the polytope that we have after $k$ iterations and by $x^k$ the current optimizer. Denote by

$$SP = \{(h, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{x \in \mathbb{R}^n : hx \leq h_0\}\}$$

(9.51)

the set of all candidates for a solution of the separation problem e.g. as defined in Chapter 7.5.4. The set $SP$ is the $h_0$-polar of the polytope $P$, see (7.13) in Chapter 7.4 where $Y$ is void because $P$ by assumption is a polytope. Ideally, we wish to find a constraint that “moves” the objective function “down” as fast as possible because we know that $P \subseteq P_k$. So we want ideally a solution to the problem

$$\min_{(h, h_0) \in SP} \max_{x \in \mathbb{R}^n} \{cx : x \in P_k \cap \{x \in \mathbb{R}^n : hx \leq h_0\}\}. \quad (9.52)$$

Since $P$ is a polytope this min-max problem has a solution if $P \neq \emptyset$ and if $P = \emptyset$ we simply declare an arbitrary “violated” inequality to be the solution, e.g. $hx = cx \leq h_0 = cx^k - 100$. If the objective function value of (9.52) is greater than or equal to $cx^k$, then by the outer inclusion principle the current iterate $x^k$ solves the linear optimization problem $\max\{cx : x \in P\}$ and we are done. Otherwise, the objective function value is less than $cx^k$ and let us call any $(h, h_0) \in \mathbb{R}^{n+1}$ that solves (9.52)

![Fig. 9.8. The outer inclusion principle in $\mathbb{R}^2$](image_url)
an optimal separator for \( P \) with respect to the objective function \( cx \) that we wish to maximize over \( P \). It follows that \( hx^k > h_0 \) and we can iterate. What we like to have ideally is not necessarily what we can do in computational practice and indeed, we are not aware of any linear optimization problem for which a solution to the min-max problem (9.52) is known. We refer to this problem sometimes as the problem of “finding the right cut” because we are evidently cutting off a portion of the polytope \( P_k \) by an optimal separator \( hx \leq h_0 \) and the cut is “right” because it moves the objective function value as much as possible. A general solution to (9.52) does not seem possible, but for certain classes of optimization problems an answer to this problem may be possible.

Since a solution to (9.52) seems elusive, we have to scale down our aspirations somewhat and approximate the problem of finding the right cut. The next best objective that comes to one’s mind is to ask for \((h, h_0) \in SP\) such that the amount of violation \( hx^k - h_0 \) is maximal. So we want to solve

\[
\max \{hx^k - h_0 : (h, h_0) \in SP, \|h\|_\infty = 1\}, \tag{9.53}
\]

where \( \|h\|_\infty = \max\{|h_j| : 1 \leq j \leq n\} \) is the \( \ell_\infty \)-norm. Because we normalized by \( \|h\|_\infty = 1 \) the maximum in (9.53) exists if \( P \neq \emptyset \). If \( P = \emptyset \), then any inequality \( hx \leq h_0 \) with \( hx^k - h_0 > 0 \) is simply declared to be a “solution” to (9.53). If the objective function value in (9.53) is less than or equal to zero, we stop: the current iterate \( x^k \) maximizes \( cx \) over \( P \). Otherwise, any solution to (9.53) is a most violated separator for \( P \).

It follows that \( hx^k > h_0 \) and we can iterate. Indeed, in all of the computational work that preceded as well as that followed the advent of the ellipsoid algorithm the constraint identification (or separation) problem was approached in this and no other way. Posing the separation problem that we need to solve iteratively the way we have done it solves the separation problem of Chapter 7.5.4: the objective function value of (9.53) provides a “proof” that all constraints of \( P \) are satisfied by the current iterate \( x^k \) of the overall iterative scheme. Of course, problem (9.53) has one nonlinear constraint, but we can get around the nonlinearity easily.

We consider in the separation step of the basic ellipsoid algorithm, i.e. in Step 1, only most violated separators. This agrees not only with computational practice, but it also alleviates certain theoretical difficulties that arise when the separation step is not treated as an optimization problem on its own.

If one merely asks for some separator as we do in the first part of point 7.5(j), then – in case that \( \dim P = k < n \) – it can happen that the separation subroutine always returns a hyperplane that is parallel to the affine hull of \( P \), see e.g. Figure 9.4, and, of course, the volumina of the ellipsoids tend to zero. Thus if the volume falls below a certain value \( V_F \), say, we can no longer conclude that \( P = \emptyset \) because – even though \( P \neq \emptyset \) – the \( n \)-dimensional volume of \( P \) is zero if \( \dim P < n \).

### 9.6 \( \varepsilon \)-Solidification of Flats, Polytopal Norms, Rounding

For any \( \varepsilon > 0 \) define the \( \varepsilon \)-solidification \( P^\infty_\varepsilon \) of a polytope \( P \subseteq \mathbb{R}^n \) with respect to the \( \ell_\infty \)-norm by

\[
P^\infty_\varepsilon = \{z \in \mathbb{R}^n : \exists x \in P \text{ such that } \|x - z\|_\infty \leq \varepsilon\}, \tag{9.54}
\]
where \( \|x - z\|_\infty = \max\{|x_j - z_j| : 1 \leq j \leq n\} \) is the \( \ell_\infty \)-norm.

**Remark 9.12** (i) For every \( \varepsilon > 0 \) and nonempty polytope \( P \subseteq \mathbb{R}^n \) the set \( P_\varepsilon^\infty \subseteq \mathbb{R}^n \) is a full dimensional polytope.

(ii) If \( h x \leq h_0 \) for all \( x \in P \), then \( h x \leq h_0 + \varepsilon \|h\|_1 \) for all \( x \in P_\varepsilon^\infty \). If \( h x \leq h_0 \) for all \( x \in P_\varepsilon^\infty \), then \( h x \leq h_0 - \varepsilon \|h\|_1 \) for all \( x \in P \), where \( \|h\|_1 = \sum_{j=1}^n |h_j| \) is the \( \ell_1 \)-norm.

(iii) Let \( H x \leq h \) be any linear description of \( P_\varepsilon^\infty \). Then \( H x \leq h - \varepsilon d \) is a linear description of \( P \) where \( d \) is the vector of the \( \ell_1 \)-norms of the rows of \( H \).

Suppose \( P \) is a rational flat. Then \( P \subseteq \{x \in \mathbb{R}^n : h x = h_0\} \) for some \( (h, h_0) \in \mathbb{R}^{n+1} \). Consequently, \( h x \leq h_0 \) and \(-h x \leq -h_0 \) for all \( x \in P \). Thus

\[
P_\varepsilon^\infty \subseteq \{x \in \mathbb{R}^n : h x \leq h_0 + \varepsilon \|h\|_1, \ -h x \leq -h_0 + \varepsilon \|h\|_1\},
\]

which corresponds to “tearing apart” the equations defining the affine hull of \( P \). Since for every \( x \in P \) the hypercube \( x + \{z \in \mathbb{R}^n : |z_j| \leq \varepsilon \text{ for } 1 \leq j \leq n\} \) is contained in \( P_\varepsilon^\infty \) and this hypercube contains \( B(x, r = \varepsilon) \) the \( n \)-dimensional volume of \( P_\varepsilon^\infty \) satisfies

\[
\text{if } P \neq \emptyset \text{ then } \text{vol}(P_\varepsilon^\infty) \geq 2^n \varepsilon^n > \frac{\varepsilon^n \pi^{n/2}}{\Gamma(1+n/2)} \text{ for all } \varepsilon > 0.
\]

Since \( \dim P_\varepsilon^\infty = n \) if \( P \neq \emptyset \) the polytope \( P_\varepsilon^\infty \) has a linear description that is unique modulo multiplication by positive scalars. Thus if \( H x \leq h \) is some linear description of \( P \) then the set \( \{x \in \mathbb{R}^n : H x \leq h + \varepsilon d\} \) contains \( P_\varepsilon^\infty \) by Remark 9.12 (ii), where \( d \) is defined in part (iii). But the containment can be proper.

Let \( x^* \in \mathbb{R}^n \) be arbitrary and suppose that we have a separation subroutine for the polytope \( P \) that finds a most violated separator \( h x \leq h_0 \). If \( h x^* \leq h_0 \), then \( x^* \in P_\varepsilon^\infty \) where \( \varepsilon > 0 \) is arbitrary. From Remark 9.12 it follows that for a point outside of \( P_\varepsilon^\infty \) there exists at least one of the representations of some facet of \( P \) that is violated by it; see the text for more detail.

**Remark 9.13** Let \( H x \leq h \) and \( \phi \geq n+1 \) be such that \( \langle h^i \rangle + \langle h_i \rangle \leq \phi \) for all rows \( (h^i, h_i) \) of \( (H h) \). Let \( x^* \in \mathbb{R}^n \) and \( (H_1 h^1), (H_2 h^2) \) be a partitioning of \( (H h) \) such that

\[
h^1 - \varepsilon d^1 \leq H_1 x^* \leq h^1 + \varepsilon d^1, \quad H_2 x^* \leq h^2 - \varepsilon d^2,
\]

where \( d^1, d^2 \) are the vectors of the \( \ell_1 \)-norms of the corresponding rows of \( H_1, H_2 \). If \( 0 \leq \varepsilon \leq 2^{-5(n+1)\phi} \) then the system \( H_1 x = h^1, H_2 x \leq h^2 \) is solvable.
In Figure 9.9 we have illustrated the $\varepsilon$-solidification of three flats in $\mathbb{R}^2$. Exercise 9.7 shows that solidification works for polyhedra and states various facts about $P_c^\infty$. It shows also that we can replace the $\ell_\infty$-norm in the definition of the $\varepsilon$-solidification of a polyhedron $P$ by the $\ell_1$-norm without changing the basic properties.

The $\ell_1$-norm and the $\ell_\infty$-norm are special polytopal norms on $\mathbb{R}^n$ in the sense that their respective “unit spheres” $\{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$ where $p \in \{1, \infty\}$ are full dimensional polytopes in $\mathbb{R}^n$. Moreover, the polytopes $\{x \in \mathbb{R}^n : \|x\|_1 \leq 1\}$ and $\{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}$ are dual polytopes in the sense that there is an one-to-one correspondence between the facets of either of them and the extreme points of the other. More generally, let $\| \cdot \|_P$ be any polytopal norm on $\mathbb{R}^n$, i.e. the unit sphere with respect to $\| \cdot \|_P$ is a polytope. Given $\| \cdot \|_P$ define for $y \in \mathbb{R}^n$ the “length” of $y$ in the dual norm $\| \cdot \|_P^*$ by

$$\|y\|_P^* = \max\{y^T x : x \in \mathbb{R}^n, \|x\|_P \leq 1\}.$$  \hfill (9.57)

The maximum exists because $\{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$ is a polytope, i.e. a compact convex subset of $\mathbb{R}^n$, and $y^T x$ is continuous in $x$. You prove that $\| \cdot \|_1$ and $\| \cdot \|_\infty$ is a pair of dual norms. Exercise 9.8 shows that $\| \cdot \|_P$ is a polytopal norm on $\mathbb{R}^n$ and that it satisfies Hölder’s inequality

$$y^T x \leq \|y\|_P^* \|x\|_P$$

for all $x, y \in \mathbb{R}^n$. \hfill (9.58)

It also shows that an $\varepsilon$-solidification of polyhedra can be defined with respect to any polytopal norm on $\mathbb{R}^n$. Moreover, if we have a separation subroutine for any polyhedron $P$ that finds a most violated separator then we can separate points from it using the corresponding $\varepsilon$-solidification of $P$.

### 9.6.1 Rational Rounding and Continued Fractions

**Rational rounding** is the process of approximating a real number $\Theta \in \mathbb{R}$ as a ratio of two integer numbers $p/q$ with $q \geq 1$. If the denominator $q = 1$ then $p = \lfloor \Theta \rfloor$ or $p = \lfloor \Theta \rfloor$ are the only “reasonable” choices with a maximum error of $1/2$. We denote by

$$[\Theta] = \min\{\Theta - \lfloor \Theta \rfloor, \lfloor \Theta \rfloor - \Theta\}$$  \hfill (9.59)

the smaller of the two fractional parts of $\Theta$ and pronounce $[\Theta]$ as “frac of $\Theta$.”

**Remark 9.14** Let $\Theta$ and $Q > 1$ be any real numbers. Then there exists an integer $q$ such that $1 \leq q \leq Q$ and $[q\Theta] \leq Q^{-1}$.

A ratio $p/D$ with integers $p$ and $D \geq 1$ is a **best approximation** to $\Theta \in \mathbb{R}$ if

$$[D\Theta] = |D\Theta - p|$$

and $[q\Theta] > |D\Theta|$ for all $1 \leq q < D$. \hfill (9.60)

Two sequences of integers $q_1 = 1 < q_2 < q_3 < \cdots$ and $p_1, p_2, p_3, \ldots$ such that $p_n/q_n$ is a best approximation to $\Theta$ for all $1 \leq q \leq D = q_n$ can be constructed inductively as follows. Initially, we let $q_1 = 1$ and find an integer $p_1$ such that $|q_1\Theta - p_1| = |\Theta| \leq 1/2$. By definition $p_1/q_1$ is a best approximation of $\Theta$ with $D = 1$. If $p_1 = q_1\Theta$, i.e. if $\Theta$ is an integer, the inductive process stops. So suppose that we have constructed the $n \geq 1$ first, best approximations to $\Theta$ and that the process does not stop. Then we have integers $p_n$ and $q_n$ such that $p_n \neq q_n\Theta$, i.e. $[q_n\Theta] > 0$. From Remark 9.14 when applied with $Q > |q_n\Theta|^{-1}$ it follows that there exist integer numbers $q \geq 1$ such
that \([q_1\theta] < [q_n\theta]\). Let \(q_{n+1}\) be the smallest integer and \(p_{n+1}\) be a corresponding integer number such that

\[
[q_{n+1}\theta] = |q_{n+1}\theta - p_{n+1}|
\] (9.61)

\[
[q_{n+1}\theta] < [q_n\theta]
\] (9.62)

\[
[q\theta] \geq [q_n\theta] \text{ for all } 1 \leq q < q_{n+1}.
\] (9.63)

Since by the inductive hypothesis \(p_n/q_n\) is a best approximation to \(\theta\) with \(D = q_n\) we have from (9.62) that \(q_{n+1} > q_n\) and from (9.63) and (9.62) that the ratio \(p_{n+1}/q_{n+1}\) is a best approximation to \(\theta\) with \(D = q_{n+1}\). Consequently, the induction works, the numbers \(p_n\) and \(q_n\) have the stated properties and either the process continues or it stops.

Several important properties of this sequence of best approximations to \(\theta\) can be established that lead to a polynomially bounded algorithm for finding best approximations to rational numbers when the denominator is bounded by a prescribed number. Since \(q_n < q_{n+1}\) and applying Remark 9.14 with \(Q = q_{n+1}\) it follows from (9.63) that

\[
q_n[q_n\theta] < q_{n+1}[q_n\theta] \leq 1.
\] (9.64)

Moreover, the signs of \(q_n\theta - p_n\) and \(q_{n+1}\theta - p_{n+1}\) alternate, i.e.

\[
(q_n\theta - p_n)(q_{n+1}\theta - p_{n+1}) \leq 0.
\] (9.65)

All numbers in the sequence are integers and

\[
p_nq_{n+1} - p_{n+1}q_n = q_n(q_{n+1}\theta - p_{n+1}) - q_{n+1}(q_n\theta - p_n),
\] (9.66)

\[
p_nq_{n+1} - p_{n+1}q_n = \pm 1,
\] (9.67)

\[
\text{sign}(p_nq_{n+1} - p_{n+1}q_n) = -\text{sign}(q_n\theta - p_n)
\] (9.68)

\[
p_nq_{n+1} - p_{n+1}q_n = -(p_{n-1}q_n - p_nq_{n-1})
\] (9.69)

for all \(p_n, q_n\) that the inductive process generates. From (9.69) \(p_n(q_{n+1} - q_{n-1}) = q_n(p_{n+1} - p_{n-1})\). From (9.67) \(\text{g.c.d.}(p_n, q_n) = 1\) and since the coprime representation of a rational is unique, there exists some positive integer \(a_n\) such that \(q_{n+1} - q_{n-1} = a_nq_n\) and \(p_{n+1} - p_{n-1} = a_np_n\). So for all \(n \geq 2\) of the inductive process there exist integers \(a_n \geq 1\) such that

\[
q_{n+1} = a_nq_n + q_{n-1}, \quad p_{n+1} = a_np_n + p_{n-1}.
\] (9.70)

Multiply the first part of (9.70) by \(\theta\), subtract the second and use the alternating signs (9.65). We get

\[
|q_n\theta - p_n| = |q_n\theta - p_n| + |q_{n+1}\theta - p_{n+1}|.
\] (9.71)

Now \(|q_{n+1}\theta - p_{n+1}| = [q_{n+1}\theta] < [q_n\theta] = |q_n\theta - p_n|\) by (9.62). Hence

\[
a_n = \left[ \frac{|q_{n-1}\theta - p_{n-1}|}{|q_n\theta - p_n|} \right],
\] (9.72)

which together with (9.70) gives a procedure to calculate \(p_{n+1}\), \(q_{n+1}\) once the values of \(p_n\) and \(q_n\) are known for \(k \leq n\) where \(n \geq 2\).
To start the procedure to calculate $p_n$ and $q_n$ iteratively from (9.70) and (9.72) we have to know what the first two iterations of the inductive process produce in terms of $p_n$ and $q_n$ or prescribe a start that is consistent with the inductive hypothesis that we have made to derive the above properties.

If $|\Theta| \geq 1$ and $\Theta$ is not an integer, then we can always write $\Theta = |\Theta| + \theta'$ with $0 < \theta' < 1$ and a best approximation to $\theta'$ yields instantaneously a best approximation to $\Theta$. So we can assume WLOG that $0 < \Theta < 1$. If $0 < \Theta \leq 1/2$ then $p_1 = 0$, $q_1 = 1$ yields a best approximation to $\Theta$ for $D = 1$ and by Exercise 9.9 (ii) $p_2 = 1$, $q_2 = |\Theta^{-1}|$ does the same for $D = |\Theta^{-1}|$. So if we initialize
\[ p_0 = 1, \quad q_0 = 0, \quad p_1 = 0, \quad q_1 = 1, \tag{9.73} \]
then formulas (9.70) and (9.72) produce precisely the respective best approximations to $\Theta$ for $n \leq 2$, the inductive hypothesis applies and so we can continue to use the formulas until the process stops, i.e. $\Theta = p_{n+1}/q_{n+1}$, if it stops at all. If $1/2 < \Theta < 1$ then the initialization (9.73) produces $p_1 = 0$, $q_1 = 1$ which is, of course, not a best approximation to $\Theta > 1/2$. Calculating we get from (9.70) and (9.72) $p_2 = 1$, $q_2 = 1$ because $a_1 = 1$, which gives a best approximation of $\Theta > 1/2$ for $D = 1$. Carrying out one more step in the iterative application of (9.70) and (9.72) with the initialization (9.73) we get $a_2 = \lfloor \Theta^{-1} \rfloor$, $p_3 = a_2$ and $q_3 = a_2 + 1$. By Exercise 9.9 (iii) $p_3/q_3$ is a best approximation to $\Theta$ for all $1 \leq q < D = \lfloor \Theta^{-1} \rfloor + 1$. Now the inductive hypothesis applies to $p_2$, $q_2$ and $p_3$, $q_3$, we ignore the first iteration and thus we can continue to use the formulas like in the first case.

If the number $\Theta$ equals $r/s$ with integers $r,s \geq 1$ and $\text{g.c.d}(r,s) = 1$, then $r/s$ is itself a best approximation to $\Theta$ for all $1 \leq q < s$, the $q_n$ are strictly increasing and thus $q_n = s$, $p_n = r$ at some point and the process stops. If $\Theta$ is irrational then $\lim_{n \to \infty} \frac{p_n}{q_n} = \Theta$ because by (9.64) we have $|\Theta - p_n/q_n| < q_n^{-2}$ and $1 = q_1 \leq q_2 < q_3 < \cdots$ for any $0 < \Theta < 1$. With the initialization (9.73) it follows from (9.65) and (9.67) that for all $n \geq 0$
\[ (-1)^{n+1}(q_n\Theta - p_n) \geq 0, \quad p_nq_{n+1} - p_{n+1}q_n = (-1)^n. \tag{9.74} \]

Consider now the best approximation problem for $\Theta \in \mathbb{R}$ relative to a prescribed integer number $D \geq 2$: we wish to find integer numbers $p$ and $1 \leq q \leq D$ such that $|\Theta - p/q|$ is as small as possible.

**Best Approximation Algorithm (\(\Theta, D\))**

**Step 0:** Set $a_0 := |\Theta|$, $\Theta := \Theta - a_0$, $p_0 := 1$, $q_0 := 0$, $p_1 := 0$, $q_1 := 1$, $n := 1$.

**Step 1:** If $q_n \Theta = p_n$ stop "$p_n/q_n$ is a best approximation".\[ \text{if } q_n > D \text{ go to Step 3. Set } a_n := \left\lfloor \frac{q_n \Theta - p_n}{q_n} \right\rfloor. \]

**Step 2:** Set $p_{n+1} := a_n p_n + p_{n-1}$, $q_{n+1} := a_n q_n + q_{n-1}$. Replace $n$ by $n$, go to Step 1.

**Step 3:** Set $k := \left\lfloor \frac{Dq_n - q_{n-1}}{q_n} \right\rfloor$, $p'_n := p_n - kq_n$, $q'_n := q_n - kq_n$. Replace $n$ by $n$. go to Step 1.

**Remark 9.15** (Correctness and finiteness) For rational $\Theta$ and integer $D \geq 2$ the best approximation algorithm’s run time is polynomial in the digital size of its input.
\[ \Theta = \Theta_1 = \frac{1}{a_1 + \Theta_2} = \frac{1}{a_1 + \frac{1}{a_2 + \Theta_3}} = \cdots = \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \cdots + \frac{1}{a_N + \Theta}}}} \]

\[ \text{Fig. 9.10. Continued fractions for a rational number} \]

To relate the preceding to the continued fraction process define

\[ \Theta_n = \frac{|q_n \Theta - p_n|}{|q_{n-1} \Theta - p_{n-1}|} \quad \text{for } n \geq 1, \]

where we assume again like in the algorithm that the integer part of the original data has been cleared away, i.e. \( 0 < \Theta < 1 \). It follows from the initialization (9.73) that \( \Theta_1 = 0 \) and \( 0 \leq \Theta_n < 1 \) for \( n \geq 1 \) from (9.62). From (9.71) we get

\[ \Theta_n^{-1} = a_n + \Theta_{n+1} \quad \text{for all } n \geq 1 \text{ with } \Theta_n > 0. \quad (9.75) \]

Now suppose that \( \Theta \) is a rational number. Then by the above \( \Theta = p_{N+1}/q_{N+1} \), say, so that \( \Theta_{N+1} = 0 \) and thus by (9.75) \( \Theta_N^{-1} = a_N \). Consequently we can write \( \Theta \) like in Figure 9.10, which explains the term “continued fraction.” If \( \Theta \) is irrational then \( \Theta_n > 0 \) for all \( n \geq 1 \) and the continued fraction goes on “forever”, which permits one to find high-precision rational approximations of irrational numbers.

Rational rounding can e.g. be used in the context of establishing the polynomiality of linear programming via the combination of the binary search algorithm 2 and the basic ellipsoid algorithm, see Chapter 9.3.1, to find the optimal objective function value \( z_P \) exactly; see the text.

### 9.7 Optimization and Separation

Here the fundamental polynomial-time equivalence of optimization and separation for rational polyhedra is established. It is shown that for any rational polyhedron \( P \subseteq \mathbb{R}^n \) the linear optimization problem

\[ \max \{ cx : x \in P \} \]

can be solved in time that is polynomial in \( n \), the facet complexity \( \phi \) of \( P \) and \( \langle c \rangle \) if and only if the separation problem (9.53) is solvable in time that is polynomial in \( n, \phi \) and \( \langle x^k \rangle \).

The geometric idea of our construction is simple and illustrated in Figures 9.11 and 9.12 for a full dimensional polyhedron in \( \mathbb{R}^2 \). In Figure 9.11 we depict the situation when \( cx \) with \( c = (-1, -1) \) is maximized, while Figure 9.12 illustrates the basic idea for finding direction vectors in the asymptotic cone of a polyhedron. For any integer \( u \geq 1 \) denote

\[ P_u = P \cap \{ x \in \mathbb{R}^n : -u \leq x_j \leq u \text{ for } 1 \leq j \leq n \}. \quad (9.76) \]
Its \( \varepsilon \)-solidification \( P^\varepsilon_u \) with respect to the \( \ell_1 \)-norm is either empty or a full dimensional polytope in \( \mathbb{R}^n \). To find direction vectors in the asymptotic cone we define

\[
\Lambda = \phi + 5n\phi + 4n^2\phi + 1.
\]  

(9.77)

Setting \( u = 2^\Lambda > 2^{4n\phi} \) it follows from point 7.5(b) that all extreme points of \( P \) are properly contained in \( P_u \). Moreover, we get a sufficiently large portion of the “unbounded” region of \( P \) by doubling this value of \( u \) which permits us to find direction vectors in the asymptotic cone of \( P \) by solving two linear optimization problems rather than one.

At the outset the polyhedron \( P \subseteq \mathbb{R}^n \) may be empty, it may have none or several optimal extreme points or the objective function may be unbounded. To encompass all possibilities, we perturb the original objective function so as to achieve uniqueness of the maximizer over the larger polytope \( P_u \) where \( u = 2^{\Lambda+1} \). To locate the unique maximizer \( x^{\text{max}} \) of the perturbed objective function over \( P_u \) where \( u \in \{2^\Lambda, 2^{\Lambda+1}\} \) we let the algorithm run until \( x^{\text{max}} \) is the only rational point in the remaining \( \varepsilon \)-optimal set with components that have denominators \( q_j \geq 1 \) and \( q_j \leq 2^{6n\phi} \). We have illustrated the basic idea in the second part of Figure 9.11 which “zooms” into the area of the first part depicted by a square and that contains \( x^{\text{max}} \). Running the algorithm “long enough” we find an \( \varepsilon \)-optimal rational vector \( x^{\text{max}} \in P^\varepsilon_u \) “close” to \( x^{\text{max}} \). Using the best approximation algorithm of Chapter 9.6 we can round componentwise to obtain \( x^{\text{max}} \), see Remark 9.16. If the maximizer \( x^{\text{max}} \) obtained this way satisfies \( |x^{\text{max}}_j| < 2^\Lambda \) for all \( 1 \leq j \leq n \), then we are done and an optimal extreme point of \( P \) for the original objective function has been located. If the maximizer \( x^{\text{max}} \) satisfies \( |x^{\text{max}}_j| = 2^\Lambda \) for some \( 1 \leq j \leq n \), then either the linear optimization problem over \( P \) has an unbounded objective function value or an optimal extreme point of \( P \) may still exist. In either case, we execute the algorithm a second time to optimize the same perturbed objective function over \( P \), where \( u = 2^{\Lambda+1} \). We get a second, unique maximizer \( y^{\text{max}} \), say, over the larger polytope and – since we have used identical objective functions – the difference vector \( y = y^{\text{max}} - x^{\text{max}} \) belongs to the asymptotic cone of the polyhedron \( P \). This will let us decide whether the objective function is unbounded or bounded and in the latter case we find an optimizing point as well.

Exercise 9.10 reviews the perturbation technique of Chapter 7.5.4 and summarizes part of what we need to establish the validity of our construction. The last part of Exercise 9.10 shows in particular that we can assume WROG that every nontrivial linear inequality \( hx \leq h_0 \) belonging to a linear description of a rational polyhedron \( P \) of facet complexity \( \phi \) satisfies \( \|h\|_\infty = 1 \).
9.7. OPTIMIZATION AND SEPARATION

9.7.1 \(\varepsilon\)-Optimal Sets and \(\varepsilon\)-Optimal Solutions

The following remark makes the first part of the construction precise and gives an analytical meaning to the terms "\(\varepsilon\)-optimal set" and "\(\varepsilon\)-optimal solution": the set \(F_\varepsilon\) defined in (9.78) is an \(\varepsilon\)-optimal set and any rational \(x \in F_\varepsilon\) is an \(\varepsilon\)-optimal solution to the linear optimization problem over \(P_u\).

**Remark 9.16** Let \(P \subseteq \mathbb{R}^n\) be a rational polytope of facet complexity \(\phi\), \(P_u\) be as defined in (9.76) with integer \(u \geq 2^{4n\phi}\) and let \(P^\varepsilon_u\) be the \(\varepsilon\)-solidification of \(P_u\) with respect to the \(\ell_1\)-norm. Let \(x^\text{max} \in P_u\) be the unique maximizer of \(dx\) over \(P_u\) and \(z_P = dx^\text{max}\), where \(d\) has rational components and \(\|d\|_\infty = 1\). Define

\[
F_\varepsilon = P^\varepsilon_u \cap \{x \in \mathbb{R}^n : dx \geq z_P - \varepsilon\},
\]

where \(0 < \varepsilon \leq 2^{-\Psi}\) and \(\Psi = 9n\phi + 12n^2\phi + \langle d \rangle\). Then \(F_\varepsilon\) is a full dimensional polytope, \(\text{vol}(F_\varepsilon) \geq 2^n \varepsilon^n / n!\) and every extreme point \(y \in F_\varepsilon\) satisfies \(|y_j - x^\text{max}_j| < 2^{-6n\phi - 1}\) for \(1 \leq j \leq n\). Moreover, rounding any rational \(\overline{x} \in F_\varepsilon\) componentwise by the best approximation algorithm with \(\Theta = \overline{x}\) and \(D = 2^{6n\phi}\) we obtain \(x^\text{max}\) and thus the maximizer \(x^\text{max}\) in time polynomial in \(n, \phi\) and \(\langle \overline{x} \rangle\).

9.7.2 Finding Direction Vectors in the Asymptotic Cone

The following remark makes the second part of the construction precise; see also Exercise 9.11 (i) below.

**Remark 9.17** Let \(P \subseteq \mathbb{R}^n\) be a rational polyhedron of facet complexity \(\phi\), let \(v > u \geq 2^{\Lambda}\) be any integers where \(\Lambda\) is defined in (9.77), let \(P_v\) and \(P_u\) be defined as in (9.76) with respect to \(v\) and \(u\), respectively, and let \(C_\infty\) be the asymptotic cone of \(P\). Then every extreme point \(x^v \in P_v\) can be written as \(x^v = x + vt\) where \(x, t \in \mathbb{R}^n\) are rational vectors, \(t \in C_\infty\), \(\langle t \rangle \leq 4n^2\phi\) and moreover,
$x^u = x + ut \in P_u$ is an extreme point of $P_u$. Likewise, if $x^v \in P_v$ is an extreme point of $P_v$ and $x^u = x + ut$, say, then $x^v = x^u + (v - u)t \in P_v$ is an extreme point of $P_v$.

### 9.7.3 A CCS Ellipsoid Algorithm

The CCS ellipsoid algorithm is a central cut, sliding objective version of the basic ellipsoid algorithm. It takes the number of variables $n$, a rational vector $d$ with $\|d\|_\infty = 1$, the facet complexity $\phi$ and $P$ as input. “$P$” is an identifier of the polyhedron over which we optimize the linear function $dx$ and used to communicate with the separation subroutine SEPAR$(x, h, h_0, \phi, P)$.

#### Exercise 9.7(vi)

We get

$$\text{Exercise 9.10 (v). It shows why we use the } \epsilon \text{-(u)} \text{solution algorithm. It takes the number of variables }$$

The CCS ellipsoid algorithm is a central cut, sliding objective version of the basic ellipsoid algorithm. It takes the number of variables $n$, a rational vector $d$ with $\|d\|_\infty = 1$, the facet complexity $\phi$ and $P$ as input. “$P$” is an identifier of the polyhedron over which we optimize the linear function $dx$ and used to communicate with the separation subroutine SEPAR$(x, h, h_0, \phi, P)$. $u$ specifies the hypercube with which we intersect $P$; see (9.76). $\epsilon$, $p$ and $T$ are the parameters for the $\epsilon$-solidification of $P$ in the $\ell_1$-norm, the required precision for the approximate calculation in terms of binary positions and the number of steps of the algorithm, respectively.

The subroutine SEPAR$(x^k, h, h_0, \phi, P)$ of Step 1 returns a most violated separator, i.e., a solution $(h, h_0)$ to (9.53). The normalization requirement $\|h\|_\infty = 1$ is no serious restriction at all; see Exercise 9.10 (v). It shows why we use the $\epsilon$-solidification of $P$, or $P$ in the $\ell_1$-norm: by Exercise 9.7(vi) we get $h^T x \leq h_0 + \epsilon \|h\|_\infty = h_0 + \epsilon$ as the corresponding inequality for $P^u$ or $P^r$ if $h^T x \leq h_0$ for all $x \in P$ and $(h, h_0)$ was returned by the separation subroutine. So, we “perturb” the feasible set like in Chapter 9.3 by adding a “small enough” $\epsilon > 0$ to the right-hand side $h_0$ of every most violated separator $(h, h_0)$ that the separation subroutine returns. $h$ is a column vector and $w_j \in \mathbb{R}^n$ is the $j$-th unit vector.

#### CCS Ellipsoid Algorithm $(n, d, \phi, P, u, \varepsilon, p, T)$

**Step 0:** Set $k := 0$, $x^0 := 0$, $F_0 := nuI_n$, $z_L := -nu\|d\| - 1$, $\tau := z_L$.

**Step 1:**

if $|x_j^k| > u + \varepsilon$ for some $j \in \{1, \ldots, n\}$ then

- set $h := w_j$ if $x_j^k > 0$, $h := -w_j$ otherwise.

else

- call SEPAR$(x^k, h, h_0, \phi, P)$.

- if $h^T x^k \leq h_0 + \varepsilon$ then

  - Set $h := -d^T$. if $dx^k > \tau$ then set $\tau := x^k$, $\tau := dx^k$.

endif

**endif**

**Step 2:** if $k = T$ go to Step 3. Set

$$x^{k+1} \approx x^k - \frac{1}{n + 1} \frac{F_k F_k^T h}{\|F_k^T h\|},$$

$$F_{k+1} \approx \frac{n + 1/12n}{\sqrt{n^2 - 1}} F_k \left( I_n - \frac{1 - \sqrt{(n - 1)/(n + 1)}}{h^T F_k F_k^T h} F_k^T h (h^T F_k) \right),$$

where $\approx$ means that componentwise the round-off error is at most $2^{-p}$.

Replace $k + 1$ by $k$ and go to Step 1.

**Step 3:** if $\tau = z_L$ stop "$z_p = -\infty$, $P$ is empty."

Round $\tau$ componentwise to the nearest rational $x^*$ such that each component of $x^*$ has a positive denominator less than or equal to $2^{6n\phi}$. stop "$x^*$ is an optimal extreme point of $P_u$."
Remark 9.18 (Correctness and finiteness) Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron of facet complexity $\phi$. Let $d \in \mathbb{R}^n$ be a rational vector with $\|d\|_{\infty} = 1$ such that $\max \{dx : x \in P\}$ has a unique maximizer $x^{max} \in P_u$ with objective function value $z_P = dx^{max}$ if $P \neq \emptyset$, where $P_u$ is defined in (9.76) and $u \geq 2^{\ln \phi}$ is any integer. If the subroutine SEPAR$([x^h, h, h_0, \phi, P])$ returns a most violated separator $hx \leq h_0$ for $x^h$ and $P$ satisfying $\|h\|_{\infty} = 1$, then the CCS ellipsoid algorithm concludes correctly that $P_u = P = \emptyset$ or it finds $x^{max}$ if it is executed with the parameters

$$
\epsilon = 2^{-\Psi}, \ T = \lceil 6n^2 \log \frac{n^2 \epsilon}{u} \rceil, \ p = 14 + n^2 + \lceil 15n \log \frac{n^2 \epsilon}{u} \rceil,
$$

where $\Psi = 9n\phi + 12n^2\phi + \langle d \rangle$.

9.7.4 Linear Optimization and Polyhedral Separation

For any polyhedron $P \subseteq \mathbb{R}^n$ with a linear description $Hx \leq h$, say, we denote by

$$
P_{\infty} = \{y \in \mathbb{R}^n : Hy \leq 0, \ |y|_{\infty} \leq 1\}
$$

the intersection of the asymptotic cone $C_{\infty}$ of $P$ with the unit sphere in the $\ell_{\infty}$-norm.

**Linear optimization problem:** Given a rational polyhedron $P \subseteq \mathbb{R}^n$ of facet complexity $\phi$ and a rational vector $c \in \mathbb{R}^n$ (i) conclude that $P$ is empty or (ii) find $x^{max} \in P$ with $cx^{max} \geq cx$ for all $x \in P$ or (iii) find $t \in P_{\infty}$ with $ct > 0$ and $ct \geq cy$ for all $y \in P_{\infty}$.

**Polyhedral separation problem:** Given a rational polyhedron $P \subseteq \mathbb{R}^n$ of facet complexity $\phi$ and a rational vector $z \in \mathbb{R}^n$ (i) conclude that $z \in P$ or (ii) find a most violated separator for $z$ and $P$, i.e., find a rational vector $(h, h_0) \in \mathbb{R}^{n+1}$ that solves the problem

$$
\max \{hz - h_0 : (h, h_0) \in SP, \ |h|_{\infty} = 1\},
$$

where $SP = \{(h, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{x \in \mathbb{R}^n : hx \leq h_0\}\}$.

Neither problem specifies the way in which the polyhedron $P \subseteq \mathbb{R}^n$ is given: all we need is the information that the polyhedron is a subset of $\mathbb{R}^n$, that its facet complexity is at most $\phi$ and some “identifier” for $P$ that permits us to communicate to some subroutine for instance.

**Remark 9.19** Let $P \subseteq \mathbb{R}^n$ be any rational polyhedron of facet complexity $\phi$. If there exists an algorithm $A$, say, that solves the polyhedral separation problem in time that is bounded by a polynomial in $n$, $\phi$ and $\langle z \rangle$, then the linear optimization problem can be solved in time that is bounded by a polynomial in $n$, $\phi$ and $\langle c \rangle$.

To outline the proof that the statement of Remark 9.19 can be reversed as well let

$$
S = \{x^1, \ldots, x^n\} \text{ and } T = \{y^1, \ldots, y^r\}
$$

be any minimal generator of $P$ and denote by $X$ the $n \times p$ matrix with columns $x^i$, by $Y$ the $n \times r$ matrix with columns $y^i$. Either $X$ or $Y$ or both may be void. The set $SP = \{(h, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{x \in \mathbb{R}^n : hx \leq h_0\}\}$ of separators for $P$ satisfies

$$
SP = \{(h, h_0) \in \mathbb{R}^{n+1} : hX - h_0g \leq 0, \ hY \leq 0\}, \quad (9.80)
$$

where $g \in \mathbb{R}^p$ is a row vector with $p$ components equal to 1. It follows that $SP$ is a polyhedral cone in $\mathbb{R}^{n+1}$ of facet complexity at most $\phi^* = 4n^2\phi + 3$. Denote by

$$
SP_{\infty} = \{(h, h_0) \in \mathbb{R}^{n+1} : hX - h_0g \leq 0, \ hY \leq 0, \ -e \leq h \leq e\}, \quad (9.81)
$$
the polyhedron in \( \mathbb{R}^{n+1} \) over which we need to maximize \( h z - h_0 \) in order to find a most violated separator for \( z \) and \( P \), where \( e \in \mathbb{R}^n \) is a row vector with \( n \) components equal to 1. The polyhedron \( S_{P_{\infty}} \) contains the halfline defined by \((0, 1) \in \mathbb{R}^n \) if \( X \) is nonvoid, it contains the line defined by \((0, \pm 1) \in \mathbb{R}^{n+1} \) if and only if \( X \) is void and every nonzero extreme point \((h, h_0) \) of \( S_{P_{\infty}} \) satisfies \( \|h\|_{\infty} = 1 \). By Remark 9.19 we can optimize the linear function \( h z - h_0 \) over \( S_{P_{\infty}} \) in polynomial time provided that the polyhedral separation problem for \( S_{P_{\infty}} \) and any rational \((f, f_0) \in \mathbb{R}^{n+1} \), say, can be solved in time that is bounded by a polynomial in \( n, \phi \) and \((f) + (f_0) \). So we need to identify the set of separators for \( S_{P_{\infty}} \). We will do so in two steps: first we identify the set \( S_{P^*} \), say, of separators for \( S_{P} \).

Since \( S_{P} \) is a polyhedral cone in \( \mathbb{R}^{n+1} \), its set \( S_{P^*} \) of separators is a subset of all halfspaces of \( \mathbb{R}^{n+1} \) that contain the origin, i.e.

\[
S_{P^*} = \{ (x, x_{n+1}) \in \mathbb{R}^{n+1} : SP \subseteq \{ (h, h_0) \in \mathbb{R}^{n+1} : h x - h_0 x_{n+1} \leq 0 \} \},
\]

because \( S_{P} \) contains the origin of \( \mathbb{R}^{n+1} \) and with every nonzero point the cone \( S_{P} \) contains the entire halfline defined by it. It follows that

\[
S_{P^*} = \{ (x, x_{n+1}) \in \mathbb{R}^{n+1} : h x - h_0 x_{n+1} \leq 0 \text{ for all } (h, h_0) \in S_{P} \}
= \{ (x, x_{n+1}) \in \mathbb{R}^{n+1} : H x - h_0 x_{n+1} \leq 0, -x_{n+1} \leq 0 \},
\]

where \( H x \leq h \) is any linear description of the polyhedron \( P \subseteq \mathbb{R}^n \). The inequality \( x_{n+1} \geq 0 \) follows because \( S_{P} \) contains the halfline noted above. If \( P \neq 0 \) and \( X \) is void, i.e. \( P \) contains lines, then we must replace \( x_{n+1} \geq 0 \) by the equation \( x_{n+1} = 0 \), because \( S_{P} \) contains the line defined by \((0, \pm 1) \in \mathbb{R}^{n+1} \) in this case. So if \( P \neq 0 \) and \( X \) is nonvoid, then the set \( S_{P^*} \) of separators for \( P \) is the homogenization of the polyhedron \( P \) – see (7.5). If \( P \neq 0 \) and \( X \) is void, then the set \( S_{P^*} \) of separators in question is simply the asymptotic cone of \( P \) – see (7.3). Define

\[
S_{P_{\infty}} = \{ (x, x_{n+1}) \in \mathbb{R}^{n+1} : H x - h_0 x_{n+1} \leq 0, -e \leq x \leq e, 0 \leq x_{n+1} \leq 1 \},
\]

where \( e \in \mathbb{R}^n \) has \( n \) components equal to 1. So the set \( S_{P_{\infty}} \) of all separators is a nonempty polytope in \( \mathbb{R}^{n+1} \), every nonzero extreme point of which has an \( \ell_{\infty} \)-norm of 1.

Consider now \((x^0, x_{n+1}^0) \neq (x^1, x_{n+1}^1) \in S_{P_{\infty}} \) with \( x^0_{n+1} > 0 \) and \( x^1_{n+1} > 0 \). Then \((x^0_{n+1} x^0, x^0_{n+1} x_{n+1}^0) \in S_{P_{\infty}} \) and \((x^0_{n+1} x^0, x^0_{n+1} x_{n+1}^0) \in S_{P_{\infty}} \). If for some \((f, f_0) \in \mathbb{R}^{n+1} \) we have

\[
f(x^0_{n+1} x^0) - f_0 x^0_{n+1} x_{n+1}^0 > f(x^0_{n+1} x^1) - f_0 x^0_{n+1} x_{n+1}^1 > 0,
\]

then the point \((x^0, x^0_{n+1}) \) is a “more violated” separator for \((f, f_0) \) and \( S_{P} \) than \((x^1, x^1_{n+1}) \), because for all \( \lambda \geq 0 \) \((x^0_{n+1} x^0, \lambda x^0_{n+1} x^1_{n+1}) \in S_{P^*}, (\lambda x^0_{n+1} x^1, \lambda x^0_{n+1} x_{n+1}^1) \in S_{P^*} \) and the previous inequalities remain true for the entire open halfline, i.e. for all \( \lambda > 0 \).

Thus even if \( f x^0 - f_0 x^0_{n+1} > f x^0 - f_0 x^0_{n+1} > 0 \), rather than \( f x^0 - f_0 x^0_{n+1} > f x^0 - f_0 x^0_{n+1} \), for the “original” points \((x^0, x^0_{n+1}), (x^1, x^1_{n+1}) \in S_{P^*} \), it can happen that \((x^0, x^0_{n+1}) \) defines a more violated separator. It follows that we have to scale the “homogenizing” components of two violated separators \((x^0, x^0_{n+1}) \) and \((x^1, x^1_{n+1}) \) with \( x^0_{n+1} > 0 \) and \( x^1_{n+1} > 0 \) to have equal values if we want to decide which one of the two is more violated than the other; see also Exercise 9.12 (iii) below.

Suppose that \((f, f_0) \in \mathbb{R}^{n+1} \) is given, that \( P \neq 0 \) and that algorithm \( B \), say, solves the linear optimization problem over \( P \subseteq \mathbb{R}^n \). We run the algorithm with the objective function vector \( e = f \).

Assume that algorithm \( B \) finds \( x_{\text{max}} \in P \) with \( f x_{\text{max}} \geq f x \) for all \( x \in P \). If \( f x_{\text{max}} \leq f_0 \) then we conclude that \((f, f_0) \in S_{P} \). Otherwise, \( f x_{\text{max}} - f_0 > 0 \) and \( f x_{\text{max}} - f_0 \geq f x - f_0 \) for all \( x \in P \).
Then

\[ x^0 = \alpha x^{\text{max}}, \quad x^0_{n+1} = \alpha \quad \text{where} \quad \alpha^{-1} = \left\| \begin{pmatrix} x^{\text{max}} \\ 1 \end{pmatrix} \right\|_\infty \]  

(9.83)

solves the polyhedral separation problem for \((f, f_0)\) and \(SP\); see the text.

Suppose that algorithm \(B\) finds \(t \in P_\infty\) with \(ft > 0\) and \(ft \geq fy\) for all \(y \in P_\infty\). Then

\[ x^0 = t, \quad x^0_{n+1} = 0 \]  

(9.84)

solves the polyhedral separation problem for \((f, f_0) \in \mathbb{R}^{n+1}\) and \(SP\); see the text.

It follows that if \(P \neq \emptyset\) we can solve the polyhedral separation problem for any rational \((f, f_0) \in \mathbb{R}^{n+1}\) and the cone \(SP\) in time that is polynomially bounded in \(n, \phi\) and \(\langle f \rangle + \langle f_0 \rangle\) by solving \(\max\{fx : x \in P\}\). If we conclude that \((f, f_0) \in SP\) define \((x^0, x^0_{n+1}) = (0, 0)\); otherwise the most violated separator \((x^0, x^0_{n+1})\) for \((f, f_0) \in \mathbb{R}^{n+1}\) and \(SP\) is given by (9.83) or (9.84), respectively.

It remains to show that we can solve the separation problem for \((f, f_0) \in \mathbb{R}^{n+1}\) and \(SP_\infty\).

The polyhedron \(SP_\infty\) differs from the cone \(SP\) by exactly \(2n\) constraints of the form \(−1 \leq h_j \leq 1\) for all \(1 \leq j \leq n\), which we can check by LIST-and-CHECK in polynomial time. Given \((x^0, x^0_{n+1})\) define

\[ \alpha = f x^0 - f_0 x^0_{n+1}, \quad \beta = \max \{0, 1 - f_j\}, \quad \gamma = \max \{0, 1 + f_j\}, \]  

where \(f_j\) is the \(j\)-th component of \(f \in \mathbb{R}^n\). If \(\max\{\alpha, \beta, \gamma\} = 0\) we conclude that \((f, f_0) \in SP_\infty\). Otherwise, if \(\max\{\alpha, \beta, \gamma\} = \alpha\) then \((x^0, x^0_{n+1})\) is a most violated separator for \((f, f_0)\) and \(SP_\infty\). If \(\max\{\alpha, \beta, \gamma\} = \beta\) then \((x^0, x^0_{n+1}) = (u^k, 1) \in \mathbb{R}^{n+1}\) is a most violated separator for \((f, f_0)\) and \(SP_\infty\) where \(k \in \{1, \ldots, n\}\) is such that \(\beta = 1 - f_k\). Likewise, if \(\max\{\alpha, \beta, \gamma\} = \gamma\) then \((x^0, x^0_{n+1}) = (-u^k, 1) \in \mathbb{R}^{n+1}\) is a most violated separator for \((f, f_0)\) and \(SP_\infty\) where \(k \in \{1, \ldots, n\}\) is such that \(\gamma = 1 + f_k\).

The preceding combination of some algorithm \(B\) for the linear optimization problem and of LIST-and-CHECK yields a separation routine \(\text{SEPAR}^\star(f, f_0, x^0, x^0_{n+1}, \phi, SP_\infty)\) that solves the polyhedral separation problem for rational \((f, f_0) \in \mathbb{R}^{n+1}\) and \(SP_\infty\) if the underlying polyhedron \(P \subseteq \mathbb{R}^n\) is nonempty. Moreover, if the running time for algorithm \(B\) is bounded by a polynomial in \(n, \phi\) and \(\langle c \rangle\), then the running time of \(\text{SEPAR}^\star\) is evidently bounded by a polynomial in \(n, \phi\) and \(\langle f \rangle + \langle f_0 \rangle\).

To complete the outline of the proof that we can solve the problem

\[ \max\{hz - h_0 : (h, h_0) \in SP_\infty\} \]

for any rational \(z \in \mathbb{R}^n\) in polynomial time if a polynomial-time algorithm \(B\) for the linear optimization problem is known, we proceed as follows.

We run the algorithm \(B\) a first time with the objective function \(c = 0\). If algorithm \(B\) concludes that \(P = \emptyset\) then we declare \(hz = z_1 > h_0 = z_1 - 1\) to be a solution to the polyhedral separation problem. Since \(P\) is empty, any inequality that is violated by \(z\) is evidently a most violated inequality for \(z\) and \(P\).

So suppose that we conclude that \(P \neq \emptyset\). Now the separation subroutine \(\text{SEPAR}^\star\) applies and thus by Remark 9.19 we can solve the linear optimization problem

\[ \max\{hz - h_0 : (h, h_0) \in SP_\infty\} \]

in time that is polynomially bounded in \(n, \phi\) and \(\langle z \rangle\).
If \((h_{\text{max}}, h_{0})\) with \(h_{\text{max}}x - h_{0} \geq hz - h_{0}\) for all \((h, h_0) \in SP_{\infty}\) is obtained, then we conclude that \(z \in P\) if \(h_{\text{max}}z \leq h_{0}\) and otherwise, a most violated separator for \(z\) and \(P\) has been obtained.

If a finite maximizer \((h_{\text{max}}, h_{0}) \in SP_{\infty}\) does not exist, then the solution to the above linear optimization problem provides a direction vector in the asymptotic cone of \(SP_{\infty}\). The polyhedron \(SP_{\infty}\) contains the halfline defined by \((0, 1) \in R^{n+1}\) along which the objective function tends to \(-\infty\). Consequently, if the unbounded case arises, then \(SP_{\infty}\) contains the line defined by \((0, \pm 1) \in R^{n+1}\) and the finite generator of \(P\) consists only of halflines, i.e. the matrix \(X\) in the definition of \(SP_{\infty}\) is void.

So we solve the linear optimization problem

\[
\max \{hz - h_0 : (h, h_0) \in SP_{\infty}, h_0 = 0\}
\]

using the separation subroutine \(\text{SEPAR}^*\), i.e. we iterate the whole procedure a second time. Now the unbounded case cannot arise and we find a new \((h_{\text{max}}, 0) \in R^{n+1}\) with \(\|h_{\text{max}}\|_{\infty} = 1\) such that \(h_{\text{max}}z \geq hz\) for all \((h, 0) \in SP_{\infty}\). If \(h_{\text{max}}z \leq 0\) then we conclude that \(z \in P\), whereas otherwise \((h_{\text{max}}, 0)\) is a most violated separator for \(z\) and \(P\) since every separator \((h, h_0)\) for \(P\) satisfies \(h_0 = 0\) in this case.

The concatenation of polynomials in some variables yields a polynomial in the same variables. Thus the entire procedure can be executed in time that is bounded by some polynomial in \(n, \phi\) and \(\langle z \rangle\).

**Remark 9.20** For any rational polyhedron \(P \subseteq R^n\) and \(n \geq 2\) the linear optimization problem and the polyhedral separation problem are polynomial-time equivalent problems.

So if either one of the two problems above is solvable in polynomial time, then so is the other. From a theoretical point of view we may thus concentrate on anyone of the two problems to study the algorithmic “tractability” of linear optimization problems over rational polyhedra \(P\) in \(R^n\).

Remark 9.19 has several important implications which we do not prove in detail. Among these are that if either problem is polynomially solvable for some rational polyhedron \(P \subseteq R^n\), then we can find

- the dimension \(\text{dim}P\) of \(P\),
- a linear description of the affine hull \(\text{aff}(P)\) of \(P\),
- a linear description of the lineality space \(L_P\) of \(P\),
- extreme points and extreme rays of \(P\) if there are any,
- facet-defining linear inequalities for \(P\), etc

in polynomial time. The latter is of particular importance for the branch-and-cut approach to combinatorial optimization problems which rely on finding (parts of) ideal descriptions of the corresponding polyhedra.
9.8 Exercises

Exercise 9.1

Show that for $n = 1$ the ellipsoids $E_k$ are intervals and that the updating formulas (9.1) and (9.2) become $x^{k+1} = x^k - \frac{1}{2} F_k \text{sign}(a)$, $F_{k+1} = F_k$ for $k \geq 0$ where $ax \leq b$ is any inequality that is violated by $x^k$ and $\text{sign}(a) = 1$ if $a \geq 0$, $-1$ otherwise. (Hint: Note that $dd^T = 1$, $I_n = 1$ and thus the terms $n - 1$ cancel.)

The inequality $\|F^{-1}(x - x^0)\| \leq 1$ in one dimension becomes $|x - x^0| \leq |F|$, since $x$, $x^0$ and $F$ are scalars. WROG we assume that $F$ is positive. Then the inequality corresponds to the interval $[x^0 - F, x^0 + F]$. Let $ax \leq b$ be a violated inequality by $x^0$. We can assume that $a = \pm 1$ (the case $a = 0$ is meaningless), since every inequality can be brought in that form. Then the vector $d$ becomes a scalar $F a / |F a|$, i.e. $d = \text{sign}(a) = \pm 1$. Thus the updating formula for the center is given by

$$x^{k+1} = x^k - \frac{1}{2} F_k \text{sign}(a).$$

Note that $dd^T$ becomes $d^2 = 1$ and $I_n$ becomes $I_1 = 1$. Then we calculate from the updating formula for $F$

$$F_{k+1} = \sqrt{\frac{n^2}{n^2 - 1}} F_k \left(1 - 1 + \sqrt{\frac{n - 1}{n + 1}}\right) = \sqrt{\frac{n^2(n - 1)}{(n - 1)(n + 1)^2}} F_k = \frac{n}{n + 1} F_k = \frac{1}{2} F_k.$$

Exercise 9.2

Let $F$, $R$ be two $m \times n$ matrices of reals.

(i) Show $\|F\| = 0$ if and only if $F = O$, $\|\alpha F\| = |\alpha| \|F\|$ for all $\alpha \in \mathbb{R}$, $\|F + R\| \leq \|F\| + \|R\|$.

(ii) Show $\|I_n\| = \sqrt{n}$ and $\|F\|^2 = \text{trace}(F F^T)$.

(iii) Show $\|F\|_2 \leq \|F\| \leq \sqrt{n} \|F\|_2$. (Hint: Use $\text{trace}(F F^T) = \sum \lambda_i$.)

(iv) For $F$ as before, $R$ of size $n \times p$, $r \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$ show

$$\|F R\| \leq \|F\| \|R\|,$$

$$\|F (I_n - \alpha r r^T)\|^2 = \|F\|^2 - \alpha (2 - \alpha \|r\|^2) \|F r\|^2.$$

(i) We have $\|F\| = \sqrt{\sum_{j=1}^{m} \sum_{i=1}^{n} (f_{ij})^2} = 0$, which is equivalent to $f_{ij} = 0$ for all $1 \leq i \leq m$ and $1 \leq j \leq n$. Thus $F = O$. 
To prove that $\|\alpha F\| = |\alpha|\|F\|$ we calculate

$$\|\alpha F\| = \alpha \sqrt{m \sum_{i=1}^{m} \sum_{j=1}^{n} (\alpha f_{ij})^2} = \alpha \sqrt{m \sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij})^2} = |\alpha|\|F\|$$

To prove that $\|F + R\| \leq \|F\| + \|R\|$ we prove the equivalent inequality (since both parts are nonnegative) $\|F + R\|^2 \leq (\|F\| + \|R\|)^2$. We calculate

$$\|F + R\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij}^2 + r_{ij}^2) + 2 \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}r_{ij} \leq \left( \sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij}^2)^2 \right)^{1/2} \left( \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{ij}^2)^2 \right)^{1/2}$$

and thus the inequality to prove becomes

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}r_{ij} \leq \left( \sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij}^2) \right)^{1/2} \left( \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{ij}^2) \right)^{1/2}$$

which follows from the Cauchy-Schwarz inequality.

(ii) The first part follows directly from the definition of the Frobenius norm:

$$\|I_n\| = \sqrt{\sum_{k=1}^{n} \sum_{j=1}^{n} \delta_{kj}} = \sqrt{\sum_{i=1}^{n} 1} = \sqrt{n},$$

where $\delta_{ij} = 1$, $\delta_{ij} = 0$. To prove the second part, let $G = FF^T$. Then $g_{kk} = \sum_{j=1}^{n} (f_{kj})^2$. Now we have

$$\text{trace}(FF^T) = \sum_{k=1}^{m} g_{kk} = \sum_{k=1}^{m} \sum_{j=1}^{n} (f_{kj}^2) = \|F\|^2.$$ 

(iii) The matrix $F^TF$ is symmetric. Thus $\text{trace}(F^TF) = \sum_{i=1}^{n} \lambda_i$ where $\lambda_i$ are the eigenvalues of $F^TF$. Since $F^TF$ is also positive semi-definite we have $\lambda_i \geq 0$ for all $1 \leq i \leq n$. Let $\Lambda = \max\{\lambda_i : 1 \leq i \leq n\}$. Then we calculate

$$\|F\|_2 = \sqrt{\Lambda} \leq \sqrt{\sum_{i=1}^{n} \lambda_i} = \|F\|$$

where we have used the second part of (ii) for the last equality. Moreover, 

$$\|F\| = \sqrt{\sum_{i=1}^{n} \lambda_i} \leq \sqrt{n\Lambda} = \sqrt{n}\|F\|.$$
9.8. EXERCISES

(iv) From the Cauchy-Schwarz inequality we have \((f^i r_j)^2 \leq \|f^i\|^2 \|r_j\|^2\), where \(f^i\) is the \(i\)-th row of \(F\) and \(r_j\) the \(j\)-th column of \(R\). Summing over \(j\) first and over \(i\) next, we get

\[
\sum_{i=1}^{m} \sum_{j=1}^{p} (f^i r_j)^2 \leq \sum_{i=1}^{m} \|f^i\|^2 \sum_{j=1}^{p} \|r_j\|^2
\]

which using \(f^i r_j = \sum_{k=1}^{n} f_k^i r^j_k\) gives

\[
\sum_{i=1}^{m} \sum_{j=1}^{p} \left(\sum_{k=1}^{n} f_k^i r^j_k\right)^2 \leq \left(\sum_{i=1}^{m} \sum_{j=1}^{n} (f_j^i)^2\right) \left(\sum_{j=1}^{p} \sum_{i=1}^{n} (r_j^i)^2\right).
\]

Using the definition of the Frobenius norm we get \(\|FR\|^2 \leq \|F\|^2 \|R\|^2\) which proves \(\|FR\| \leq \|F\| \|R\|\), since \(\| \cdot \| \geq 0\). To prove the second relation, we calculate

\[
\begin{align*}
\|F(I_n - \alpha rr^T)\|^2 &= \text{trace}(F(I_n - \alpha rr^T)(I_n - \alpha rr^T)F^T) = \text{trace}((F - \alpha Frr^T)(F^T - \alpha rr^T F^T)) \\
&= \text{trace}(FF^T - \alpha Frr^T F - \alpha Frr^T F^T + \alpha^2 Frr^T rr^T F^T) \\
&= \text{trace}(FF^T) - 2\alpha \|Fr\|^2 + \alpha^2 \|r\|^2 \|Fr\|^2 = \|F\|^2 - \alpha(2 - \|r\|^2) \|Fr\|^2.
\end{align*}
\]

Exercise 9.3

Let \(Q = FF^T\) and \(Q_p = F_p F_p^T\), where

\[
F_p = \sqrt{\frac{n^2}{n^2 - 1}} F \left( I_n - \left(1 - \sqrt{\frac{n-1}{n+1}} dd^T\right) \right).
\]

Show that

\[
\begin{align*}
Q_p &= \frac{n^2}{n^2 - 1} Q \left( I_n - \frac{2}{n - 1} aa^T Qa \right), \\
Q_p^{-1} &= \frac{n^2 - 1}{n^2} \left( Q^{-1} + \frac{2}{n - 1} aa^T Qa \right).
\end{align*}
\]

For \(Q_p\) we calculate using (9.13)

\[
Q_p = F_p F_p^T = \frac{n^2}{n-1} F \left( I_n - \left(1 - \sqrt{\frac{n-1}{n+1}} dd^T\right) \right) \left( I_n - \left(1 - \sqrt{\frac{n-1}{n+1}} dd^T\right) \right) F^T
\]

\[
= \frac{n^2}{n^2 - 1} \left[ FF^T - 2 \left(1 - \sqrt{\frac{n-1}{n+1}} \right) Fdd^T F^T + \left(1 - \sqrt{\frac{n-1}{n+1}}\right)^2 Fdd^T dd^T F^T \right]
\]

\[
= \frac{n^2}{n^2 - 1} \left[ FF^T - 2 \left(1 - \sqrt{\frac{n-1}{n+1}} \right) Fdd^T F^T + \left(1 - \sqrt{\frac{n-1}{n+1}}\right)^2 Fdd^T dd^T F^T \right]
\]
or factoring out the scalar \( d^T d = \| d \|^2 = 1 \) from the last term
\[
= \frac{n^2}{n^2 - 1} \left[ Q - \left( 2 - 2 \sqrt{\frac{n-1}{n+1}} - 1 - \frac{n-1}{n+1} + 2 \sqrt{\frac{n-1}{n+1}} \right) Fdd^T F^T \right]
\]
and since \( Fdd^T F = \frac{Qaa^T Q}{a^T Qa} \) because \( d = F^T a / \| F^T a \| \) and \( \| d \| = 1 
\[
= \frac{n^2}{n^2 - 1} Q \left( I_n - \frac{2}{n+1} \frac{aa^T Q}{a^T Qa} \right).
\]

For \( Q^{-1}_p \) we calculate using the formula (9.22) for \( F^{-1}_p \)
\[
Q^{-1}_p = (F^{-1}_p)^T F^{-1}_p = \frac{n-1}{n^2} (F^{-1})^T \left( I_n - \left( 1 - \sqrt{\frac{n-1}{n-1}} dd^T \right) \right) \left( I_n - \left( 1 - \sqrt{\frac{n+1}{n-1}} dd^T \right) \right) F^{-1}
\]
\[
= \frac{n^2 - 1}{n^2} \left( F^{-1})^T F^{-1} - 2 \left( 1 - \sqrt{\frac{n+1}{n-1}} \right) (F^{-1})^T dd^T F^{-1} + \left( 1 - \sqrt{\frac{n+1}{n-1}} \right)^2 (F^{-1})^T dd^T dd^T F^{-1} \right)
\]
or factoring out the scalar \( d^T d = \| d \|^2 = 1 \) from the last term
\[
= \frac{n^2 - 1}{n^2} \frac{1}{Q^{-1}} \left( I_n - \frac{2}{n+1} \frac{aa^T}{a^T Qa} \right)
\]
where we have used \( \| d \| = 1 \) and \( d^T F^{-1} = a^T / \| F^T a \| \) which follows from the definition of \( d \); see (9.12).

**Exercise 9.4**

Let \( Q_k = F_k^T F_k \) be the positive definite matrix that defines the ellipsoid \( E_k \). Denote by \( \lambda_{min}^k \) the smallest and by \( \lambda_{max}^k \) the largest eigenvalue of \( Q_k \). Prove that \( \lambda_{min}^k \leq R^{2-k/2n^2} \) and \( \lambda_{max}^k \geq R^{2k-2k/n^2} \) for all \( k \) of the basic ellipsoid algorithm. (Hint: Use (9.35).)

Since \( Q_k \) is positive definite and nonsingular we have \( \lambda_{min}^k > 0 \). From \( Q_k = F_k^T F_k \) we have that the eigenvalues of \( F_k \) are \( |\mu_i| = \sqrt{\lambda_i} \) and thus \( |det F| = \prod_{i=1}^{n} \sqrt{\lambda_i} \). It follows that
\[
(\lambda_{min}^k)^{n/2} \leq |det F| \leq (\lambda_{max}^k)^{n/2}.
\]
From (9.35) we have
\[
(\lambda_{min}^k)^{n/2} \leq |det F| \leq R^{n^2 - \frac{k}{m^2}} \Rightarrow \lambda_{min}^k \leq R^{2 - \frac{k}{2n^2}}
\]
and similarly

$$(\lambda_{\text{max}}^k)^{n/2} \geq |\det \mathbf{F}| \geq R^n 2^{-\frac{k}{n}} \Rightarrow \lambda_{\text{max}}^k \geq R^2 2^{-\frac{2k}{n}}.$$

---

**Exercise 9.5**

(i) Write a computer program of the DCS ellipsoid algorithm in a computer language of your choice for the linear programming problem (LP).

(ii) Solve the problems of Exercises 5.1, 5.9 and 8.2(ii) using your computer program.

---

(i) The following listing is an implementation of the algorithm in MATLAB. The required input is as in the simplex algorithm; see Exercise 5.2.

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This is the implementation of the DCS Ellipsoid algorithm
%% as found on pages 309-310.
%%
%% NAME : dcsel
%% PURPOSE: Solve the LP: \( \max \{cx: a\mathbf{x} \leq b, \mathbf{x} \geq 0\} \)
%% INPUT : The matrix \( a \) and the vectors \( c \) and \( b \).
%% OUTPUT : \( \mathbf{z} \) : the optimal value
%% \( \mathbf{x} \) : the optimal solution
%% \( k \) : the number of iterations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[m,n]=size(A);
A = [A;-eye(n)];
b = [b zeros(1,n)];
[m,n]=size(A);
z1=-10000;
zu=10000;
eps=10^(-9);
R=500;
Vf=10^(-6);
R0=sqrt(n)*(1+R/2);
veps=eps*ones(m,1);
z=z1+1;
z0=z;
x = (R/2)*ones(n,1);
H=R0*eye(n);
```

---
\textbf{Ellipsoid Algorithms}

\begin{verbatim}
\texttt{k=0;}
\texttt{f0=(1+1/n)^(-(n+1)/2) * (1-1/n)^(-(n-1)/2);} \n\texttt{V=R0^n * pi^(n/2) / gamma(1+n/2);} \\
\texttt{while (1<2)} \\
\texttt{\hspace{1em}slack=b'+veps- A*x ;} \\
\texttt{\hspace{1em}[mxv,j]=min(slack);} \\
\texttt{\hspace{1em}if (mxv >= 0)} \\
\texttt{\hspace{2em}xstar=x+H*H'*c'/norm(c*H);} \\
\texttt{\hspace{2em}help=A*(xstar-x);} \\
\texttt{\hspace{1em}if (all(help <= 0)), error('Unbounded'), end;} \\
\texttt{\hspace{1em}lambda=10000;} \\
\texttt{\hspace{2em}for i=1:m,} \\
\texttt{\hspace{3em}if (help(i) > 0)} \\
\texttt{\hspace{4em}if (slack(i) /help(i) <= lambda)} \\
\texttt{\hspace{5em}lambda=slack(i)/help(i);} \\
\texttt{\hspace{3em}end;} \\
\texttt{\hspace{2em}end;} \\
\texttt{\hspace{1em}if (lambda >= 1), error('Unbounded'), end;} \\
\texttt{\hspace{1em}if (c*(x+lambda*(xstar-x)) > z)} \\
\texttt{\hspace{2em}xbar=x+lambda*(xstar-x);} \\
\texttt{\hspace{2em}z=c*xbar;} \\
\texttt{\hspace{1em}end;} \\
\texttt{\hspace{1em}if ((c*x > z & z0 > zl+1) | (mxv <0 & z0-z > mxv))} \\
\texttt{\hspace{2em}theta=b(j)+eps;} \\
\texttt{\hspace{2em}alpha=(A(j,:)*x-theta)/norm(A(j,:)*H);} \\
\texttt{\hspace{2em}r=A(j,:)';} \\
\texttt{\hspace{2em}else} \\
\texttt{\hspace{3em}alpha=(z-c*x)/norm(c*H);} \\
\texttt{\hspace{3em}r=-c'}; \\
\texttt{\hspace{3em}z0=z;} \\
\texttt{\hspace{2em}end;} \\
\texttt{\hspace{1em}if ((alpha >=1 | V < Vf) & (z < zu & z > zl))} \\
\texttt{\hspace{2em}fprintf('Optimal solution found in %d iterations, Vol= %8.4f\n',k-1,V);} \\
\texttt{\hspace{2em}fprintf('%8.4f',xbar);} \\
\texttt{\hspace{2em}fprintf('\n,z);} \\
\texttt{\hspace{2em}return;} \\
\texttt{\hspace{1em}elseif (z <= zl)} \\
\texttt{\hspace{2em}fprintf('Infeasible');} \\
\texttt{\hspace{2em}return;} \\
\texttt{\hspace{1em}elseif (z >= zu)} \\
\texttt{\hspace{2em}fprintf('Unbounded');} \\
\texttt{\hspace{2em}return;} \\
\texttt{\hspace{1em}end;} \\
\end{verbatim}
xold=x;
x=x-(1+n*alpha)*H'*r/((n+1)*norm(H'*r));
quan1=sqrt(((n-1)*(1-alpha))/((n+1)*(1+alpha)));
quan2=n*sqrt((1-alpha^2)/(n^2-1));
H=quan2*H*(eye(n)-(1-quan1)*H'*r'/(H'*r)^2);
V=(1-alpha)*f0*V*(1-alpha^2)^((n-1)/2);
fprintf('%10.5f ',x);
fprintf(' cx=%10.5f, z=%10.5f
',c*x,z);
z1=c*(x-xold);
help1=A*(x-xold);
if (all(help1 < 0) & z1 > 0), error('Unbounded'), end;
slack1=b'+veps-A*xold;
mumax=-10000;
mumin=10000;
substep=1;
for i=1:m,
    if (help1(i) > 0 )
        if (slack1(i)/help1(i) <=mumin)
            mumin=slack1(i)/help1(i);
        end;
    elseif (help1(i) < 0)
        if (slack1(i)/help1(i) >= mumax)
            mumax=slack1(i)/help1(i);
        end;
    else
        if (slack(i) < 0)
            substep=0;
        end;
    end;
end;
if (mumin < mumax & substep > 0) 
    substep=0;
end;
if (substep > 0 )
    if (z1 < 0)
        mubar=mumax;
    else
        mubar=mumin;
    end;
    if (c*xold+mubar*z1 > z)
        xbar=xold+mubar*(x-xold);
        z=c*xbar;
    end;
end;
k=k+1;
end;
(ii) For the data of Exercise 5.1 we get

```matlab
>> clear
>> psdat
>> dcsel
-4.03134 165.32289 -88.70846 80.64577 cx= 294.36367, z=-9999.00000
-10.02109 30.06256 -30.06291 10.02074 cx= -30.06467, z=-9999.00000
-52.62810 42.08574 12.86073 -5.27118 cx= 61.90156, z=-9999.00000
9.55652 35.42522 -3.34090 -46.70324 cx= 18.61864, z=-9999.00000
3.54364 -1.94823 -14.64155 9.78702 cx= -37.74956, z=-9999.00000
-8.02842 7.99811 2.99137 -2.24117 cx= 15.42061, z=-9999.00000
9.55652 35.42522 -3.34090 -46.70324 cx= 18.61864, z=-9999.00000
3.54364 -1.94823 -14.64155 9.78702 cx= -37.74956, z=-9999.00000

Optimal solution found in 119 iterations, Vol= 0.0000
z= 30.6572
```

For Exercise 5.9 with \( n = 3 \), \( a = b = 2 \) and \( c = 5 \), we get

```matlab
>> clear
>> psdat
>> dcsel
-85.49883 82.25058 208.06265 cx= 30.56848, z=-9999.00000
11.12853 -64.29617 171.42596 cx= 87.34772, z=-9999.00000
1.20391 12.52250 126.70039 cx= 156.60104, z=-9999.00000
-10.39740 2.76268 74.50551 cx= 38.41428, z=-9999.00000
1.90067 -6.57926 24.50713 cx= 18.95129, z=-9999.00000
1.38161 3.95126 -6.60592 cx= 6.82304, z=-9999.00000
0.58555 2.21549 17.22113 cx= 23.99432, z=-9999.00000
-0.03957 0.85243 9.95152 cx= 11.49810, z=-9999.00000
-0.53385 -0.22534 29.23529 cx= 26.64921, z= 19.92770
1.33403 -1.70743 24.00886 cx= 25.93012, z= 19.92770
```
## 9.8. Exercises

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>cx=</th>
<th>z=</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.98365</td>
<td>-3.93039</td>
<td>26.10890</td>
<td>22.18273</td>
<td>19.92770</td>
</tr>
<tr>
<td>0.15010</td>
<td>1.90863</td>
<td>13.56085</td>
<td>17.97850</td>
<td>19.92770</td>
</tr>
<tr>
<td>-0.29774</td>
<td>1.49264</td>
<td>22.51259</td>
<td>24.30692</td>
<td>20.09956</td>
</tr>
<tr>
<td>-0.92891</td>
<td>0.90634</td>
<td>24.64934</td>
<td>22.74638</td>
<td>20.09956</td>
</tr>
<tr>
<td>0.71750</td>
<td>-0.20129</td>
<td>18.03809</td>
<td>20.50551</td>
<td>20.09956</td>
</tr>
<tr>
<td>0.55850</td>
<td>-0.93332</td>
<td>23.94083</td>
<td>24.30821</td>
<td>21.48207</td>
</tr>
<tr>
<td>0.29328</td>
<td>1.24644</td>
<td>19.78129</td>
<td>23.44728</td>
<td>21.48207</td>
</tr>
<tr>
<td>0</td>
<td>0.00120</td>
<td>25.00568</td>
<td>25.00008</td>
<td>24.99545</td>
</tr>
<tr>
<td>0</td>
<td>0.00141</td>
<td>-0.00352</td>
<td>25.00029</td>
<td>24.99888</td>
</tr>
<tr>
<td>0</td>
<td>0.00057</td>
<td>0.00211</td>
<td>24.99013</td>
<td>24.99661</td>
</tr>
<tr>
<td>0</td>
<td>-0.00020</td>
<td>0.00127</td>
<td>24.99280</td>
<td>24.99452</td>
</tr>
</tbody>
</table>

Optimal solution found in 68 iterations, Vol= 0.0000

0.0002 0.0017 24.9913

z= 24.9957

For the data of Exercise 8.2(ii) we get

```matlab
>> clear
>> psdat
dcsel
```

194.74376 26.28120 cx= 26.28120, z=-9999.00000

-0.42637 25.28755 cx= -25.28755, z=-9999.00000

-20.12962 -1.44852 cx= 1.44852, z= -14.24470

8.55908 8.48696 cx= 8.48696, z= 0.00000

14.71900 -5.87632 cx= -5.87632, z= 0.00000

6.88480 -1.39700 cx= -1.39700, z= 0.00000

14.14111 5.56176 cx= 5.56176, z= 0.00000

6.40835 -1.85392 cx= 1.85392, z= 0.00000

8.98593 0.61797 cx= -0.61797, z= 0.00000

8.12674 -0.20599 cx= 0.20599, z= 0.00000

8.41314 0.06866 cx= -0.06866, z= 0.00000

8.31767 -0.02289 cx= 0.02289, z= 0.00000

8.34949 0.00763 cx= -0.00763, z= 0.00000

8.33889 -0.00254 cx= 0.00254, z= 0.00000

8.34242 0.00085 cx= -0.00085, z= 0.00000

8.34124 -0.00028 cx= 0.00028, z= 0.00000

8.34164 0.00009 cx= -0.00009, z= 0.00000

8.34151 -0.00003 cx= 0.00003, z= 0.00000

8.34155 0.00001 cx= -0.00001, z= 0.00000

8.34153 -0.00000 cx= 0.00000, z= 0.00000

8.34154 0.00000 cx= -0.00000, z= 0.00000

8.34154 -0.00000 cx= 0.00000, z= 0.00000

8.34154 0.00000 cx= -0.00000, z= 0.00000
Exercise 9.6

(i) Let $H_{k+1}$ be as defined in (9.45), i.e.,

$$
H_{k+1} = n\sqrt{\frac{1 - \alpha_k^2}{n^2 - 1}} H_k \left( I_n - \left( n - \frac{(n-1)(1 - \alpha_k)}{(n+1)(1 + \alpha_k)} \right) \frac{H_k^T r (r^T H_k)}{\|H_k^T r\|^2} \right).
$$

Like we proved (9.20) show that

$$
\det H_{k+1} = \left( 1 + \frac{1}{n} \right)^{-\frac{n+1}{2}} \left( 1 - \frac{1}{n} \right)^{-\frac{n-1}{2}} (1 - \alpha_k^2)^{n-1} (1 - \alpha_k) \det H_k .
$$

(ii) Define $G_k = H_k H_k^T$ and show using (9.45) that

$$
G_{k+1} = \frac{n^2(1 - \alpha_k^2)}{n^2 - 1} G_k \left( I_n - \frac{2(1 + n\alpha_k)}{(n+1)(1 + \alpha_k)} r r^T G_k r \right),
$$

$$
G_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} (G_k^{-1} + \frac{2(1 + n\alpha_k)}{(n-1)(1 - \alpha_k)} r r^T G_k r).
$$

(iii) Show $\Gamma(1 + n/2) = (n/2)!$ for all even $n \geq 1$, $\Gamma(1 + n/2) = \frac{n! \sqrt{\pi}}{[\frac{n}{2}]! 2^{n}}$ for all odd $n \geq 1$.

---

(i) Letting $\alpha = 1 - \sqrt{\frac{(n-1)(1 - \alpha_k)}{(n+1)(1 + \alpha_k)}}$ the proof of (9.20) applies unchanged since $0 \leq \alpha_k < 1$. Thus replacing $F_p$ by $H_{k+1}$, $F$ by $H_k$ and $d$ by $H_k^T r / \|H_k^T r\|$, since $\|d\| = 1$ we get

$$
\det H_{k+1} = \left( \frac{n^2(1 - \alpha_k^2)}{n^2 - 1} \right)^{n/2} \frac{(n-1)(1 - \alpha_k)}{(n+1)(1 + \alpha_k)} \det H_k .
$$

To bring this to the required form we have

$$
\sqrt{\frac{(n-1)(1 - \alpha_k)}{(n+1)(1 + \alpha_k)}} = \sqrt{\frac{(n-1)n(1 - \alpha_k)^2}{(n+1)n(1 - \alpha_k^2)}} = \left( 1 - \frac{1}{n} \right)^{1/2} \left( 1 + \frac{1}{n} \right)^{-1/2} (1 - \alpha_k)(1 - \alpha_k^2)^{-1/2}
$$
and
\[ \frac{n^2}{n^2 - 1} (1 - \alpha_k^2) = \frac{n}{n + 1} \frac{n}{n - 1} (1 - \alpha_k^2) = \left(1 + \frac{1}{n}\right)^{-1} \left(1 - \frac{1}{n}\right)^{-1} (1 - \alpha_k^2). \]

Thus after grouping of terms we get
\[ \det H_{k+1} = \left(1 + \frac{1}{n}\right)^{-2} \left(1 - \frac{1}{n}\right)^{-1} (1 - \alpha_k^2) (1 - \alpha_k) \det H_k. \]

(ii) Using the definition of \( H_{k+1} \) we calculate:
\[
G_{k+1} = H_{k+1}H_k^T = n^2 \frac{1 - \alpha_k^2}{n^2 - 1} \left( H_k - \left(1 - \sqrt{(n-1)(1-\alpha_k)} \right) \frac{H_kH_k^T r^T H_k}{\|H_k^T r\|^2} \right)
\]
\[
= n^2 \frac{1 - \alpha_k^2}{n^2 - 1} \left[ H_kH_k^T - \frac{H_kH_k^T r^T H_k}{\|H_k^T r\|^2} \left(1 - \sqrt{(n-1)(1-\alpha_k)} \right) \left(\frac{H_kH_k^T r^T H_k}{\|H_k^T r\|^2}\right) \right]
\]
and factoring out the term \( r^T H_k H_k^T r = \|H_k^T r\|^2 \) from the numerator of the last fraction
\[
= n^2 \frac{1 - \alpha_k^2}{n^2 - 1} \left( G_k - \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} G_k r^T G_k \right) = n^2 \frac{1 - \alpha_k^2}{n^2 - 1} G_k \left( I_n - \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} r^T G_k r \right)
\]

For the inverse \( G_{k+1}^{-1} \) we have
\[
G_{k+1}^{-1} = n^2 \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} \left( I_n - \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} r^T G_k r \right)^{-1} G_k^{-1}.
\]

Using Exercise 4.1 (ii) with \( u = -\frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} r \) and \( v = G_k r \) we calculate \( v^T u \neq -1 \) because \( \alpha_k \neq 1 \) and thus
\[
\left( I_n - \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} r^T G_k \right)^{-1} = I_n + \frac{1}{1 - \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} r^T G_k r} \frac{2(1 + n\alpha_k)}{(n + 1)(1 + \alpha_k)} r^T G_k r
\]
\[
= I_n + \frac{2(1 + n\alpha_k)}{n + n\alpha_k + 1 + \alpha_k - 2 - 2n\alpha_k} r^T G_k r
\]
\[
= I_n + \frac{2(1 + n\alpha_k)}{(n - 1)(1 - \alpha_k)} r^T G_k r.
\]
It follows that
\[ G_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left( G_k^{-1} + \frac{2(1 + n\alpha_k)}{(n-1)(1 - \alpha_k)} \frac{rr^T}{G_k r} \right) . \]

(iii) For \( n > 1 \) and even, i.e. \( n = 2k \) with \( k \geq 1 \) is integer we have
\[ \Gamma(1 + n/2) = \Gamma(1 + k) = k\Gamma(k) = k! = \left( \frac{n}{2} \right)! \]
where we have used elementary properties of the gamma function; see Chapter 7.7. For \( n \geq 1 \) and odd, i.e. \( n = 2k + 1 \) where \( k \geq 0 \) is integer we have
\[ \Gamma\left( \frac{2k + 1}{2} \right) = \Gamma\left( \frac{2k - 1}{2} + 1 \right) = \frac{2k - 1}{2} \Gamma\left( \frac{2k - 1}{2} \right) , \]
where the last equality follows from the fact that \( 2k - 1 \) is even. Thus we get
\[ \Gamma\left( 1 + \frac{n}{2} \right) = \Gamma\left( \frac{2k + 3}{2} \right) = \frac{1}{2^{k+1}} \Gamma\left( \frac{1}{2} \right) \prod_{\ell=0}^{k} (2\ell + 1) . \]
Using the identity \( \prod_{\ell=0}^{k} (2\ell + 1) = \frac{(2k+1)!}{k!2^k} \) and that \( \Gamma\left( \frac{1}{2} \right) = \sqrt{\pi} \) we get
\[ \Gamma\left( 1 + \frac{n}{2} \right) = \frac{n!}{\left( \frac{n}{2} \right)!} 2^n \sqrt{\pi} . \]

Exercise 9.7
Let \( P \subseteq \mathbb{R}^n \) be a polyhedron and define \( P^n_\varepsilon = \{ z \in \mathbb{R}^n : \exists x \in P \text{ such that } \|x - z\|_\infty \leq \varepsilon \} . \)

(i) Show that \( P^n_\varepsilon \) is a polyhedron in \( \mathbb{R}^n \) and that \( P \neq \emptyset \) implies \( \dim P^n_\varepsilon = n \) for \( \varepsilon > 0 \).

(ii) Show that if \( z \in P^n_\varepsilon \) is an extreme point then there exists an extreme point \( x \in P \) such that \( z_i - x_i = \pm \varepsilon \) for all \( 1 \leq i \leq n \).

(iii) Let \( y \in \mathbb{R}^n \). Show that \( x + (y) \in P^n_\varepsilon \) for some \( x \in P^n_\varepsilon \) if and only if there exists \( \tilde{x} \in P \) such that \( \tilde{x} + (y) \in P \). Show that the asymptotic cone \( C_\infty \) of \( P \) is the asymptotic cone of \( P^n_\varepsilon \).

(iv) Show that \( P^n_\varepsilon \) has at most \( p2^n \) extreme points where \( p \) is the number of extreme points of \( P \).

(v) Suppose that the facet complexity of \( P \) is \( \phi \) and its vertex complexity is \( \nu \). Show that for rational \( \varepsilon \geq 0 \) the polyhedron \( P^n_\varepsilon \) has a vertex complexity of \( 2\nu + 2n(\varepsilon) \) and a facet complexity of \( 3\phi + 2(\varepsilon) \).

(vi) Define \( P^1_\varepsilon \) by
\[ P^1_\varepsilon = \{ z \in \mathbb{R}^n : \exists x \in P \text{ such that } \|x - z\|_1 \leq \varepsilon \} . \]
Show that \( P^1_\varepsilon \) is a polyhedron, \( hx \leq h_0 + \varepsilon \|h\|_\infty \) for all \( x \in P^1_\varepsilon \) if \( hx \leq h_0 \) for all \( x \in P \) and \( hx \leq h_0 - \varepsilon \|h\|_\infty \) for all \( x \in P \) if \( hx \leq h_0 \) for all \( x \in P \). (Hint: Use that \( \sum_{j=1}^n |x_j| \leq \varepsilon \) if and only if \( \sum_{j=1}^n \delta_j x_j \leq \varepsilon \) for the \( 2^n \) vectors \( (\delta_1, \ldots, \delta_n) \) with \( \delta_j \in \{ +1, -1 \} \) for all \( 1 \leq j \leq n \).)
(viii) Show that if \( z \in P_{\epsilon}^1 \) is an extreme point then \( z = x \pm \epsilon u^i \) where \( x \in P \) is an extreme point of \( P \) and \( u^i \in \mathbb{R}^n \) is the \( i \)-th unit vector for some \( 1 \leq i \leq n \). Show that \( P_{\epsilon}^1 \) has at most \( 2n^p \) extreme points if \( P \) has \( p \) extreme points.

(ix) Suppose \( P \) has a facet complexity of \( \phi \) and a vertex complexity of \( \nu \). Show that \( P_{\epsilon}^1 \) has vertex complexity of \( 2\nu + 2(\epsilon) \) and a facet complexity of \( 3\phi + 2(\epsilon) \).

(x) Prove that

\[
\text{if } P \neq \emptyset \text{ then } \text{vol}(P_{\epsilon}^1) \geq \frac{2^n\epsilon^n}{n!} > \frac{\epsilon^n n^{n/2}}{n^n (1 + n/2)} \text{ for all } \epsilon > 0.
\]

(Hint: Use that \( P_{\epsilon}^1 \supseteq x + \{z \in \mathbb{R}^n : \|z\|_1 \leq 1\} \) and \( P_{\epsilon}^1 \supseteq B(x, r = \epsilon/n) \) for every \( x \in P \).

(x) Show that Remark 9.13 remains correct if we replace the \( \ell_1 \)-norm by the \( \ell_{\infty} \)-norm.

(i) Let \( S = \{x^1, \ldots, x^p\} \), \( T = \{r^1, \ldots, r^t\} \) be any finite generator of \( P \). Then every \( x \in P \) can be written as \( x = X\mu + R\nu \) where \( X = (x^1, \ldots, x^p) \) is an \( n \times p \) matrix, \( R = (r^1, \ldots, r^t) \) an \( n \times t \) matrix, \( \mu \geq 0, f_\mu = 1, \nu \geq 0 \) and \( f = (1, \ldots, 1) \in \mathbb{R}^p \). If \( P \) is pointed, then \( x^1, \ldots, x^p \) are the extreme points and \( r^1, \ldots, r^t \) are the direction vectors of the extreme rays of \( P \). Let \( e^2 = (1, \ldots, 1) \in \mathbb{R}^n \). Then \( P_{\epsilon}^\infty \) is the image of the polyhedron

\[
PP_{\epsilon} = \{(z, \mu, \nu) \in \mathbb{R}^{n+p+t} : -\epsilon e \leq X\mu + R\nu - z \leq \epsilon e, f_\mu = 1, \mu \geq 0, \nu \geq 0\}
\]

under the linear transformation with the matrix \( L = (I_n, 0, 0) \), i.e., \( P_{\epsilon}^\infty \) is the image of \( PP_{\epsilon} \) when we project out the variables \( \mu \) and \( \nu \). By point 7.3(g) it follows that \( P_{\epsilon}^\infty \) is a polyhedron. To prove the dimension of \( P_{\epsilon}^\infty \), suppose that \( P \neq \emptyset \) and let \( x \in P \). Then the points \( x, x + \epsilon u^1, \ldots, x + \epsilon u^n \) are \( n+1 \) affinely independent points in \( P_{\epsilon}^\infty \) where \( u^i \) is the \( i \)-th unit vector in \( \mathbb{R}^n \). Thus \( \text{dim} P_{\epsilon}^\infty = n \).

(ii) Let \( z \in P_{\epsilon}^\infty \) and \( x \in P \) be such that \( z_i - x_i = \pm \epsilon \) for all \( 1 \leq i \leq n \) and suppose that \( z \) is an extreme point of \( P_{\epsilon}^\infty \) but \( x \) is not an extreme point of \( P \). It follows that \( x = \mu x^1 + (1 - \mu)x^2 \) where \( 0 < \mu < 1 \), \( x^1, x^2 \in P \) and \( x^1 \neq x^2, x^1 \neq x \neq x^2 \). Then we get \( z = x \pm \epsilon e = \mu x^1 + (1 - \mu)x^2 \pm (\mu + 1 - \mu)e = \mu(x^1 \pm \epsilon e) + (1 - \mu)(x^2 \pm \epsilon e) \). From the definition of \( P_{\epsilon}^\infty \), it follows that \( z^1 = x^1 \pm \epsilon e \in P_{\epsilon}^\infty \) and \( z^2 = x^2 \pm \epsilon e \in P_{\epsilon}^\infty \). Moreover, \( z^1 \neq z^2 \) and thus \( z = \mu z^1 + (1 - \mu)z^2 \) with \( 0 < \mu < 1 \) contradicts the assumption that \( z \) is an extreme point of \( P_{\epsilon}^\infty \).

(iii) From Remark 9.12 (ii) we have that the asymptotic cones of \( P \) and \( P_{\epsilon}^\infty \) are the same, since every valid inequality \( hx \leq h_0 \) for one of the polyhedra gives rise to a valid inequality of the form \( hx \leq h_0 \) for the other and vice versa. An inequality is valid for \( P \), if it is satisfied by every \( x \in P \). To prove the first part suppose that \( (y) \in P_{\epsilon}^\infty \) is a halfline of \( P_{\epsilon}^\infty \), i.e., \( hy \leq 0 \) for all \( h \) such that \( hx \leq h_0 \) is valid for \( P_{\epsilon}^\infty \). By part (ii) of Remark 9.12 we have that \( hx \leq h_0 - \epsilon \|h\|_1 \) is valid for \( P \) for all such \( h \). Then \( hy \leq 0 \) for all \( h \) such that \( hx \leq h_0 - \epsilon \|h\|_1 \) is valid for \( P_{\epsilon}^\infty \). On the other hand, suppose that \( y \) is not a halfline of \( P \), i.e., there exists a valid inequality \( hx \leq h_0 \) for \( P_{\epsilon}^\infty \) such that for some \( x \in P \) there exists \( \alpha > 0 \) such that \( h(x + \alpha y) > h_0 - \epsilon \|h\|_1 \). Then \( ahy > 0 \) for some \( \alpha > 0 \), i.e., \( hy > 0 \) which contradicts the assumption that \( (y) \) is a halfline of \( P_{\epsilon}^\infty \). On the other hand, suppose that \( (y) \) is not a halfline of \( P_{\epsilon}^\infty \). Then \( hy \leq 0 \) for all \( h \) such that \( hx \leq h_0 \) is valid for \( P \). By point (ii) of Remark 9.12 we have that \( hx \leq h_0 + \epsilon \|h\|_1 \) for all \( x \in P_{\epsilon}^1 \) for all such \( (h, h_0) \). Suppose that \( (y) \) is not a halfline of \( P_{\epsilon}^1 \). Then there exists a valid inequality \( hx \leq h_0 \) for \( P_{\epsilon}^\infty \) such that for some \( x \in P_{\epsilon}^\infty \) there exists \( \alpha > 0 \) such that \( h(x + \alpha y) > h_0 + \epsilon \|h\|_1 \). Consequently
we have \( h_0 \leq h_0 + \varepsilon \|h\|_1 < h(x + \alpha y) \leq h_0 + \alpha h y \), i.e. \( ah y > 0 \) for some \( \alpha > 0 \) which contradicts the assumption that \( y \) is a halfline of \( P \).

**(iv)** From part (ii) we have that for each extreme point \( x \) of \( P \) there exists an extreme point \( z \) of \( P_\varepsilon^x \) such that \( z_i = x_i \pm \varepsilon \), for \( 1 \leq i \leq n \). That is each component of \( x \) can be either increased or decreased by \( \varepsilon \). So for each extreme point of \( P \) there are at most \( 2^n \) different points \( z \) that can be constructed as above. Thus if \( P \) has \( r \) extreme points, then \( P_\varepsilon^x \) has at most \( r2^n \) extreme points.

**(v)** From Remark 9.12 (ii) we have that \( h z \leq h_0 + \varepsilon \) is a valid inequality for \( P_\varepsilon^x \) if \( h x \leq h_0 \) is valid for \( P \). Thus for the facet complexity of \( P_\varepsilon^x \) we calculate

\[
(h) + (\langle h_0 + \varepsilon \|h\|_1 \rangle) = (h) + 2\langle h_0 \rangle + 2\varepsilon + 2\langle \|h\|_1 \rangle \leq 3(h) + 2\langle h_0 \rangle + 2\varepsilon + 2(1 - \varepsilon) \leq 3(h) + 2\varepsilon + 2\varepsilon 
\]

where we have used \( \langle \|h\|_1 \rangle \leq (h) - n + 1 \); see Chapter 7.5. For the vertex complexity, from part (ii) of this exercise we have that if \( z \) is an extreme point of \( P_\varepsilon^x \) then \( z_i = x_i \pm \varepsilon \) for \( 1 \leq i \leq n \) where \( x \) is an extreme point of \( P \). We calculate

\[
\langle z \rangle = \sum_{i=1}^{n} \langle z_i \rangle = \sum_{i=1}^{n} (x_i \pm \varepsilon) \leq 2 \sum_{i=1}^{n} (\langle x_i \rangle + \varepsilon) \leq 2 \sum_{i=1}^{n} \langle x_i \rangle + 2n\varepsilon = 2\langle x \rangle + 2n\varepsilon \leq 2\langle x \rangle + 2n\varepsilon .
\]

**(vi)** Let \( \Delta = (\sigma_k^2)_{k=1}^{2^n} \) be the matrix with rows the \( 2^n \) vectors with components \( \pm 1 \). The constraint \( \|x - z\|_1 \leq \varepsilon \) is written as \( \|x - z\|_1 = \sum_{i=1}^{n} |x_i - z_i| \leq \varepsilon \) and it is equivalent to the constraints \( \Delta(x - z) \leq \varepsilon e \); see also Exercise 2.2(ii). So we have that

\[
P_\varepsilon^{x_1} = \{ z \in \mathbb{R}^n : \exists x \in P \text{ such that } \Delta(x - z) \leq \varepsilon e \} .
\]

Since \( P \) is a polyhedron, let like in part (i) \( X, R \) be the matrices corresponding to a finite generator of \( P \). Then

\[
x = X\mu + R\nu \text{ where } \mu \geq 0, \ f\mu = 1, \ \nu \geq 0
\]

and \( f = (1, \ldots, 1)^T \in \mathbb{R}^{p} \). \( P_\varepsilon^{x_1} \) is the image of the polyhedron

\[
PP_\varepsilon^{x_1} = \{ (z, \mu, \nu) \in \mathbb{R}^{n+p+t} : \Delta(X\mu + R\nu - z) \leq \varepsilon e, \ f\mu = 1, \ \mu \geq 0, \ \nu \geq 0 \}
\]

under the linear transformation \( L = (I_n \ O_p \ O_t) \) where \( O_p \) is \( n \times p \) and \( O_t \) is \( n \times t \), i.e. when we project out \( \mu \) and \( \nu \). It follows by point 7.3(g) that \( P_\varepsilon^{x_1} \) is a polyhedron.

Suppose now that \( z \in P_\varepsilon^{x_1} \). Then there exists \( x \in P \) such that \( \sum_{i=1}^{n} |x_i - z_i| \leq \varepsilon \). We calculate

\[
hz - hx = h(x - z) \leq \sum_{i=1}^{n} |h_i| |z_i - x_i| \leq \sum_{i=1}^{n} \|h\|_\infty |z_i - x_i| \leq \varepsilon \|h\|_\infty
\]

Since for every \( x \in P \) we have \( hx \leq h_0 \) it follows that \( hz \leq h_0 + \varepsilon \|h\|_\infty \) for all \( z \in P_\varepsilon^{x_1} \). On the other hand, suppose that \( hz \leq h_0 \) for all \( z \in P_\varepsilon^{x_1} \) but there exists \( x \in P \) such that \( hx > h_0 - \varepsilon \|h\|_\infty \). Let \( z \) be such that \( z_i = x_i \) for all \( i \neq k \) and \( z_k = x_k + \varepsilon \) if \( h_k \geq 0 \), \( z_k = x_k - \varepsilon \) if \( h_k < 0 \), where \( 1 \leq k \leq n \) is such that \( |h_k| = \|h\|_\infty \) and \( 1 \leq i \leq n \). Thus if \( h_k \geq 0 \) we have

\[
hz = hx + h_k\varepsilon = hx + \varepsilon \|h\|_\infty > h_0 - \varepsilon \|h\|_\infty + \varepsilon \|h\|_\infty = h_0
\]

and if \( h_k < 0 \) that

\[
hz = hx - h_k\varepsilon = hx + \varepsilon \|h\|_\infty > h_0 - \varepsilon \|h\|_\infty + \varepsilon \|h\|_\infty = h_0
\]
i.e., \(hz > h_0\) for some \(z \in P\) which is a contradiction.

\(\textbf{vii}\) Suppose that \(z = x \pm \varepsilon u^i\) is an extreme point of \(P^1_\varepsilon\) but \(x \in P\) is not an extreme point of \(P\). Then there exist \(0 < \mu < 1\) and \(x^1 \neq x^2 \neq x^1\) such that \(x = \mu x^1 + (1 - \mu)x^2\). But then we have

\[
z = x \pm \varepsilon u^i = \mu x^1 + (1 - \mu)x^2 \pm \varepsilon u^i = \mu x^1 + (1 - \mu)x^2 \pm \varepsilon u^i \pm (1 - \mu)\varepsilon u^i
\]

where \(z^k = x^k \pm \varepsilon u^i\) for \(k = 1, 2\). From the definition of \(P^1_\varepsilon\) it follows that \(z^1, z^2 \in P^1_\varepsilon\) and since \(z^1 \neq z^2\), because \(x^1 \neq x^2\), we have that \(z\) is the convex combination of two points in \(P^1_\varepsilon\) which contradicts the assumption that \(z\) is an extreme point.

For each extreme point \(x \in P\) there are \(2^n\) distinct points of the form \(z = x \pm \varepsilon u^i\). Thus if \(P\) has \(p\) extreme points, \(P^1_\varepsilon\) has at most \(2np\) extreme points.

\(\textbf{ix}\) From the previous part we have that the extreme points of \(P^1_\varepsilon\) are of the form \(z = x \pm \varepsilon u^i\) where \(x\) is an extreme point of \(P\). Since the vertex complexity of \(P\) is \((x) \leq \nu\), we compute

\[
\langle z \rangle = \langle x \pm \varepsilon u^i \rangle = \left(\sum_{j=1}^{n} x_j \pm \varepsilon \right) \leq 2 \sum_{j=1}^{n} \langle x_j \rangle + 2\varepsilon = 2\langle x \rangle + 2\varepsilon \leq 2\nu + 2\varepsilon.
\]

For the facet complexity we have that for any valid inequality \(hz \leq h_0 + \varepsilon \|h\|_\infty\) we have \((h) + \langle h_0 \rangle \leq \nu\) and thus we compute

\[
\langle h \rangle + \langle h_0 + \varepsilon \|h\|_\infty \rangle \leq \langle h \rangle + 2\langle h_0 \rangle + 2\varepsilon + 2\|h\|_\infty \leq 2\langle h \rangle + 2\langle h_0 \rangle + 2\varepsilon + 2\|h\|_\infty \\
\leq 2\langle h \rangle + 2\langle h_0 \rangle + 2\varepsilon + \langle h \rangle \leq 3\langle h \rangle + 2\langle h_0 \rangle + 2\varepsilon \leq 3\nu + 2\varepsilon.
\]

\(\textbf{ix}\) Since \(P^1_\varepsilon \supseteq x + \{z \in \mathbb{R}^n : \|z\| \leq 1\}\) and \(P^1_\varepsilon \supseteq B(x, \varepsilon/n)\) we have like in (9.55)

\[
\text{vol}(P^1_\varepsilon) \geq \text{vol}(B^1) > \text{vol}(B(x, \varepsilon/n)) = \frac{\pi^{n/2}\varepsilon^n}{\Gamma(1 + n/2)n^n}.
\]

Thus, we have to show that the volume of the sphere in \(\ell_1\)-norm, \(B^\varepsilon\) is given by \(2^n\varepsilon^n/n!\). To this end, it suffices to show that \(V_n = \text{vol}(B^1) = 2^n/n!\). We have

\[
V_n = \int_{-1}^{1} V_{n-1}(1 - |x_n|)dx_n = \int_{-1}^{1} f_{n-1}(1 - |x_n|)^{n-1}dx_n = f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \int_{-1}^{1} |x_n|^j dx_n
\]

\[
= f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \frac{2}{j + 1} = 2f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \frac{1}{j!(n-1-j)!} \frac{1}{j + 1}
\]

\[
= 2f_{n-1} \sum_{j=0}^{n-1} (-1)^j \frac{n!}{j!(n-1-j)!} \frac{1}{j + 1} = 2f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n}{j} \frac{1}{j + 1} = 2f_{n-1} n \sum_{\ell=1}^{n} (-1)^{\ell-1} \binom{n}{\ell}
\]

\[
= 2f_{n-1} n \sum_{\ell=1}^{n} (-1)^{\ell} \binom{n}{\ell} = 2f_{n-1} n \left(\sum_{\ell=0}^{n} (-1)^{\ell} \binom{n}{\ell} - (-1)^0 \binom{n}{0}\right)
\]

\[
= -2f_{n-1} n (1 - 1)^n - 1 = 2f_{n-1} n.
\]
So we have \( f_n = \frac{2}{n} f_{n-1} \) and thus \( f_n = 2^n / n! \).

**Exercise 9.8**

Let \( \| \cdot \|_p \) be a polytopal norm on \( \mathbb{R}^n \), i.e., \( \| \cdot \|_p \) is a norm in \( \mathbb{R}^n \) and its “unit sphere” \( B_P = \{ x \in \mathbb{R}^n : \| x \|_p \leq 1 \} \) is a polytope.

(i) Show that the “dual norm” \( \| y \|_p^* = \max \{ y^T x : \| x \|_p \leq 1 \} \) is a norm on \( \mathbb{R}^n \).

(ii) Let \( \tilde{a}^i x \leq b_i \) for \( 1 \leq i \leq m \) be any linear description of \( B_P = \{ x \in \mathbb{R}^n : \| x \|_p \leq 1 \} \) and \( a^i = \tilde{a}^i / \| \tilde{a}^i \|_p \), for \( 1 \leq i \leq m \). Show that \( B_{\tilde{P}} = \{ x \in \mathbb{R}^n : a^i x \leq 1, 1 \leq i \leq m \} \), that \( 0 \in \text{relint} B_{\tilde{P}} \) and \( \dim B_{\tilde{P}} = n \).

(iii) Show that \( \| \cdot \|_p \) is a polytopal norm on \( \mathbb{R}^n \).

(iv) Prove Hölder’s inequality \( y^T x \leq \| y \|_p \| x \|_p \).

(v) Let \( P_\varepsilon^P \) be defined by \( P_\varepsilon^P = \{ z \in \mathbb{R}^n : \exists x \in P \text{ such that } \| z - x \|_p \leq \varepsilon \} \). Show that \( P_\varepsilon^P \) is a polyhedron. Show that \( h x \leq h_0 + \varepsilon \| h \|_p \) for all \( x \in P_\varepsilon^P \) if \( h x \leq h_0 \) for all \( x \in P \). Show that \( h x \leq h_0 - \varepsilon \| h \|_p \) for all \( x \in P \) if \( h x \leq h_0 \) for all \( x \in P_\varepsilon^P \).

**(i)** For any \( y \in \mathbb{R}^n \), let \( y' = y / \| y \|_p \). Then \( y \in \mathbb{R}^n \), by the homogeneity of \( \| \cdot \|_p \), \( \| y' \|_p \leq 1 \) and \( \| y' \|_p \leq 1 \) and thus from the definition of \( \| \cdot \|_p \) we have that \( \| y' \|_p \geq y^T y' = y^T y / \| y \|_p = \| y \|^p / \| y \|_p \) where \( \| y \| \) is the Euclidean norm of \( y \), i.e., \( \| y \|_p^* \) is the ratio of two nonnegative numbers and thus it is nonnegative which is zero if and only if \( \| y \| = 0 \), i.e., if and only if \( y = 0 \). The homogeneity follows trivially since \( \| ay \|_p^* = \max \{ ay^T x : x \in \mathbb{R}^n, \| x \|_p \leq 1 \} = \alpha \max \{ y^T x : x \in \mathbb{R}^n, \| x \|_p \leq 1 \} = \alpha \| y \|_p^* \) for all \( y \in \mathbb{R}^n \) and \( \alpha \geq 0 \). Finally we have \( \| y + z \|_p^* = \max \{ (y + z)^T x : x \in \mathbb{R}^n, \| x \|_p \leq 1 \} \leq \max \{ y^T x : x \in \mathbb{R}^n, \| x \|_p \leq 1 \} + \max \{ z^T x : x \in \mathbb{R}^n, \| x \|_p \leq 1 \} = \| y \|_p^* + \| z \|_p^* \), i.e., the triangle inequality holds and thus \( \| \cdot \|_p \) is a norm.

**(ii)** We are given that \( B_P = \{ x \in \mathbb{R}^n : \tilde{a}^i x \leq b_i \text{ for } i = 1, \ldots, m \} \). From the definition (9.57) of the dual norm we have \( \| a^i \|_p^* = \max \{ a^i x : x \in B_P \} \leq b_i \) and thus \( \tilde{a}^i x \leq \| a^i \|_p^* \) for all \( x \in B_P \) and all \( 1 \leq i \leq m \). Since \( \| \cdot \|_p \geq 0 \) and \( \tilde{a}^i \neq 0 \) we have that \( \| a^i \|_p^* > 0 \) and thus dividing by \( \| a^i \|_p \) we get \( a^i x \leq 1 \) for all \( x \in B_P \) and \( 1 \leq i \leq m \). Since \( a^0 < 1 \) for \( 1 \leq i \leq m \), \( 0 \in \text{relint} B_P \). Let \( \| a \| = \max \{ \| a^i \| : 1 \leq i \leq m \} \) and \( \varepsilon = 1 / \| a \| \). By the Cauchy-Schwarz inequality, \( a^i x \leq \| a^i \| / \| a \| \leq 1 \) for \( 1 \leq i \leq m \) and for all \( x \in \mathbb{R}^n \) with \( \| x \| \leq \varepsilon \). Consequently, \( B_P \supseteq B(0, r = \varepsilon) \), i.e., \( B_P \) contains a ball with center \( 0 \in \mathbb{R}^n \) and radius \( r = \varepsilon > 0 \), and thus \( \dim B_P = n \).

**(iii)** Since \( \| \cdot \|_p \) is a polytopal norm the set \( B_P = \{ x \in \mathbb{R}^n : \| x \|_p \leq 1 \} \) is a polytope with \( \dim B_P = n \) and \( 0 \in \text{relint} B_P \). Let \( S_P = \{ x^1, \ldots, x^p \} \) be the set of the extreme points of \( B_P \) and \( Ax \leq b \) be a linear description of \( B_P \). Then we have that \( \| y \|_p^* = \max \{ y^T x : Ax \leq b \} = y^T x' \) where \( x' \in S_P \).
Thus $B_p = \{ y \in \mathbb{R}^n : \| y \|_p \leq 1 \} = \{ y \in \mathbb{R}^n : y^T x^i \leq 1 \text{ for all } 1 \leq i \leq p \}$. Since $\dim B_p = n$ and $0 \in \text{relint}B_p$ it follows that there exists $\mu \in \mathbb{R}^n, \mu \geq 0$ such that $\sum_{i=1}^p \mu_i x^i = \pm u^i$ where $u^i$ is the $i$-th unit vector in $\mathbb{R}^n$ and $1 \leq i \leq n$. Consequently, by the duality theorem of linear programming $0 \leq \max \{ y^T (\pm u^i) : y^T x^i \leq 1 \text{ for } i = 1, \ldots, p \} = \min \{ e^T \mu : \sum_{i=1}^p \mu_i x^i = \pm u^i, \mu \geq 0 \} < \infty$ for $1 \leq i \leq n$ and thus $B_p$ is a bounded set of $\mathbb{R}^n$, i.e., the “unit sphere” in the $\| \cdot \|_p$-norm is a polytope.

**Exercise 9.9**

Let $0 < \Theta < 1$.

(i) Show that $\Theta \leq |\Theta^{-1}|^{-1} < 2\Theta$ and $\Theta |\Theta^{-1}| > 1 - \Theta$.

(ii) Show that $|\Theta^{-1}|^{-1}$ is a best approximation to $\Theta$ for all $1 \leq q < |\Theta^{-1}|$.

(iii) Suppose $1/2 < \Theta < 1$. Show that $r/s$ where $r = \lfloor \frac{\Theta}{1-q} \rfloor$ and $s = \lfloor \frac{\Theta}{1-q} \rfloor + 1$ is a best approximation to $\Theta$ for all $1 \leq q < s$.

(iv) Show that $q_n \geq 2(n-1)^{1/2}$ for $n \geq 2$ for the integers $q_n$ generated by the inductive process.

(v) Suppose $0 < \Theta \neq \Theta' < 1$ and that the integers $p_n, q_n$ and $p'_n, q'_n$ generated by the respective inductive processes are such that $p_n = p'_n$ and $q_n = q'_n$ for $1 \leq n \leq N$, say. Show that $|\Theta - \Theta'| \leq 2^{-N+1}$.

(i) From the definition of the lower integer part of a number we have $|\Theta^{-1}| \leq \Theta^{-1}$ and since $0 < \Theta < 1$, we get $\Theta \leq |\Theta^{-1}|^{-1}$ and the left part of the first inequality follows. For the right part we have: $\Theta^{-1} < |\Theta^{-1}|^{-1} + 1 \leq |\Theta^{-1}|^{-1} + |\Theta^{-1}|^{-1} = 2|\Theta^{-1}|^{-1}$ and thus $|\Theta^{-1}|^{-1} < 2\Theta$, where we have used the definition of the lower integer part of a number in the first inequality and the inequality $|\Theta^{-1}|^{-1} \geq 1$ in the second. To prove that $\Theta |\Theta^{-1}|^{-1} > 1 - \Theta$ we have from the definition of the lower integer part of a number that $\Theta^{-1} < |\Theta^{-1}| + 1$. Multiplying both sides by $\Theta > 0$ we get $1 < \Theta |\Theta^{-1}| + \Theta$ and thus $\Theta |\Theta^{-1}| > 1 - \Theta$.

(ii) Applying the definition (9.60) of the best approximation with $p = 1$ and $D = |\Theta^{-1}|$ we have to show that (a) $|\Theta |\Theta^{-1}| = |\Theta| |\Theta^{-1}| - 1$ and (b) $|q\Theta| > |\Theta |\Theta^{-1}|$ for all $1 \leq q < |\Theta^{-1}|$. From the
first inequality of part (i) we have that \( \frac{1}{2} < \Theta(\Theta^{-1}) \leq 1 \) and thus \( |\Theta(\Theta^{-1})| = |\Theta(\Theta^{-1})| - \Theta(\Theta^{-1}) < \Theta(\Theta^{-1}) - |\Theta(\Theta^{-1})| \), i.e. \( |\Theta(\Theta^{-1})| = 1 - \Theta(\Theta^{-1}) \) and (a) follows. To prove (b) we have to show that \( [q\Theta] > 1 - \Theta(\Theta^{-1}) \) for all \( 1 \leq q < |\Theta^{-1}| \). First we note that \( q\Theta < |\Theta^{-1}|\Theta \) for all \( 1 \leq q < |\Theta^{-1}| \) since \( \Theta > 0 \). Thus, in particular, \( 0 < q\Theta < 1 \) and \( |q\Theta| = 0 \). If \( |q\Theta| = 1 - q\Theta \) then we have \( |q\Theta| = 1 - q\Theta > 1 - \Theta(\Theta^{-1}) \). If \( |q\Theta| = q\Theta \), we have \( [q\Theta] = q\Theta > 1 - \Theta(\Theta^{-1}) \) where we have used the second inequality of part (i) and thus (b) follows, and the proof of (ii) is complete.

(iii) To show that \( \frac{s}{r} \) with \( r = \lfloor \frac{\Theta}{1-\Theta} \rfloor \) and \( s = \lfloor \frac{\Theta}{1-\Theta} \rfloor + 1 \) is a best approximation to \( \Theta \) for all \( 1 \leq q < s \) if \( 1/2 < \Theta < 1 \) we have to prove

(a) \( s|\Theta| = |s|\Theta - r \), and (b) \( |q|\Theta > s|\Theta| \) for all integer \( q \) with \( 1 \leq q < s \).

We observe first, from \( \frac{\Theta}{1-\Theta} \geq \frac{|\Theta|}{1-\Theta} \) and \( 1 - \Theta > 0 \), that \( s|\Theta| \geq \frac{|\Theta|}{1-\Theta} \) and thus \( s|\Theta| \geq \frac{|\Theta|}{1-\Theta} \). On the other hand, \( s|\Theta| < s = r + 1 \) since \( \Theta < 1 \) and thus \( s|\Theta| = r \). To prove (a) we have to show that \( \min\{s|\Theta| - |s|\Theta, |s|\Theta - s\Theta\} = s\Theta - |s|\Theta \). This is equivalent to \( s|\Theta| - \{s|\Theta| - s\Theta\} \Theta = 1 \), which is true if \( s\Theta \) is integer, and so we can assume that \( s\Theta \) is not integer. Thus we need to show \( s|\Theta| \leq |s|\Theta + 1/2 \), i.e., \( (1 + \lfloor \frac{|\Theta|}{1-\Theta} \rfloor) \Theta \leq \lfloor \frac{|\Theta|}{1-\Theta} \rfloor + 1/2 \). Let \( x = \frac{\Theta}{1-\Theta} \). Then \( \Theta = x \) and the assertion reads \( (1 + \lfloor x \rfloor) x \leq x + 1/2 \) or equivalently, \( x \leq 2 \lfloor x \rfloor + 1 \) which is trivially true for all \( x \geq 0 \). Consequently, part (a) follows.

To prove part (b), we first prove

(b1) \( q\Theta - |q\Theta| > s|\Theta| - r \) for all integer \( q \) with \( 1 \leq q \leq r \).

We claim \( q\Theta - |q\Theta| \geq (q+1)|\Theta| - ((q+1)|\Theta| \) or equivalently, \( (q+1)|\Theta| \geq |q|\Theta + \Theta \) for all integer \( q \in [1, r] \). Since \( q \leq \lfloor \frac{\Theta}{1-\Theta} \rfloor \) we have \( q \leq \frac{\Theta}{1-\Theta} \) and thus \( q \leq (q+1)|\Theta| \). Since \( q \) is integer we get \( q \leq (q+1)|\Theta| \) and thus \( (q+1)|\Theta| \geq q|\Theta| + |\Theta| \) since \( 0 < \Theta < 1 \) if \( |q\Theta| > 0 \). If \( |q\Theta| = 0 \) then \( q = 1 \) and the assertion is true as well because \( \Theta > 1/2 \). Consequently, the claim follows and thus, it suffices to prove (b1) for \( q = r \), i.e., \( r\Theta - |r\Theta| > (r+1)|\Theta| - r \) because \( s = r + 1 \). But \( r > \Theta + |r| \Theta \) is trivially true since \( r \geq 1 \) is integer and \( 0 < \Theta < 1 \). Hence (b1) follows. We are left with proving

(b2) \( |q\Theta| - q\Theta > s|\Theta| - r \) for all integer \( q \) with \( 1 \leq q \leq r \).

Suppose first that \( \lfloor \frac{\Theta}{1-\Theta} \rfloor = \frac{\Theta}{1-\Theta} \). Then \( s|\Theta| - r = 0 \) and assume(b2) is wrong. Let \( q \in [1, r] \) be the smallest integer with \( |q\Theta| - q\Theta = 0 \). It follows that \( \Theta = p/q \) where \( 1 \leq p = |q\Theta| < q \). But \( \Theta = \frac{p}{q} \) and thus \( p(r+1) = rq \). Hence we get \( \frac{p}{q} = \frac{r}{r+1} = \frac{q}{q+p} \), i.e., both \( p \) and \( q \) are divisible by the integer \( q - p \). If \( q - p = 1 \), then \( p = r \) and \( q = r + 1 \). Otherwise, \( q - p \geq 2 \) contradicts our assumption that \( q \) is the smallest integer with the required property. Consequently, such \( q \) does not exist in the range \( 1, \ldots, r \) and (b2) follows. Now suppose \( \lfloor \frac{\Theta}{1-\Theta} \rfloor \) is not integer. Like in the proof of part (b1) we conclude \( q \leq (q+1)|\Theta| \). If \( (q+1)|\Theta| \) is integer, then we get \( q = (q+1)|\Theta| \), i.e., \( q = \frac{\Theta}{1-\Theta} \), which contradicts the assumption that \( \frac{\Theta}{1-\Theta} \) is not integer. Consequently, \( (q+1)|\Theta| \) is not integer. Thus \( |\Theta| \leq |\Theta|(q+1)|\Theta| \leq (q+1)|\Theta| \) and hence from \( 0 < \Theta < 1 \), \( (q+1)|\Theta| - (q+1)|\Theta| \geq |\Theta| - q\Theta \) for all integer \( q \in [1, r] \). Consequently, it suffices to prove (b2) for \( q = 1 \), i.e., \( 1 - \Theta > (1 + \lfloor \frac{\Theta}{1-\Theta} \rfloor) \Theta - (1 + \lfloor \frac{\Theta}{1-\Theta} \rfloor) \). Using \( x = \frac{\Theta}{1-\Theta} \), the assertion is equivalent to \( 1 + [x] > x \) which is trivially true because \( x \) is a positive noninteger. Thus (b2) follows in this case as well, and hence the proof of part (iii) is complete.

(iv) We prove the assertion by induction. From the construction of the sequence of integers \( q_i \) we have \( q_1 = 1 < q_2 < \cdots \). Thus since \( q_2 > 1 \) and integer, \( q_2 \geq 2 \geq 2^{1/2} \) and the assertion is
true for \( n = 2 \). Suppose that for \( n = k \geq 2 \) we have \( q_k \geq 2^{(k-1)/2} \). By (9.70) \( q_{k+1} = a_kq_k + q_{k-1} \geq q_k + q_{k-1} \geq 2^{(k-1)/2} + 2^{(k-2)/2} = 2^{(k-2)/2} \left( 2^{1/2} + 1 \right) > 2^{(k-2)/2} 2 = 2^k \) and thus the assertion follows for \( n = k + 1 \) and the inductive proof is complete. (Note that here we use the inductive process without the particular initialization (9.73), which "shifts" the index \( n \) of the inductive process by 1 if \( 1 > \Theta > 1/2 \).

(v) Assume WLOG that \( 0 < \Theta < \Theta' < 1 \) and that the inductive process carries out at least \( N \geq 1 \) iterations. By (9.65), i.e., because the signs of \( q_n \Theta - p_n \) and \( q_n \Theta' - p_n' \) alternate, we have either \( \Theta' \leq p_N/q_N \) or \( p_N/q_N \leq \Theta \). Since \( \Theta' \neq \Theta \), the process continues for at least one more iteration for either \( \Theta \) or \( \Theta' \) or both. Suppose that \( \Theta' \leq p_N/q_N \). Then \( p_{N+1}/q_{N+1} = \Theta \) and 0 < \( \Theta' - \Theta \leq \frac{p_N - p_{N+1}}{q_N - q_{N+1}} = \frac{1}{q_{N+1} N} \leq 2^{-N+\frac{3}{2}}, \) by part (iv). Suppose that \( p_N/q_N \leq \Theta \). Then \( p_{N+1}/q_{N+1} \geq \Theta' \) and 0 < \( \Theta' - \Theta \leq \frac{p_N - p_{N+1}}{q_N - q_{N+1}} = \frac{1}{q_{N+1} N} \leq 2^{-N+\frac{3}{2}} \) by part (iv) as well because \( p_N = p_N' \) and \( q_N = q_N' \) by assumption. Thus \( |\Theta - \Theta'| \leq 2^{-N+\frac{3}{2}} \leq 2^{-N+1} \) as we have asserted.

Exercise 9.10

Let \( P \subseteq \mathbb{R}^n \) be a rational polyhedron of facet complexity \( \phi \). Let \( P_u = P \cap \{ x \in \mathbb{R}^n : -u \leq x_j \leq u \text{ for } 1 \leq j \leq n \} \) for some integer \( u \geq 1 \) and let \( e \in \mathbb{R}^n \) be any rational vector.

(i) Every extreme point \( x \in P_u \) is a rational vector with components \( x_j = p_j/q_j \) with integers \( 0 \leq |p_j| < u^2 \phi + 1 \) and \( 1 \leq q_j < 2^{u \phi} \) for \( 1 \leq j \leq n \).

(ii) Any two extreme points \( x, y \in P_u \) with \( cx > cy \) satisfy \( cx > cy + 2^{-12n^2 \phi - (e)} \).

(iii) For \( \Delta \geq 1 + u 2^{(u \phi + 12n^2 \phi + (e))} + 1 \) let \( d_j = \Delta^n c_j + \Delta^{n-j} \) for \( 1 \leq j \leq n \) and \( d = (d_1, \ldots, d_n) \). Then the linear optimization problem \( \max \{ dx : x \in P_u \} \) has a unique maximizer \( x^{\text{max}} \in P_u \) and \( cx^{\text{max}} = \max \{ cx : x \in P_u \} \).

(iv) Define \( d = \tilde{d}/\|\tilde{d}\|_\infty \) where \( \tilde{d} \) is defined in part (iii). Then \( \langle d \rangle \leq 3.5n(n-1)[\log_2 \Delta] + 2(n-1)\langle e \rangle \) and thus for \( u = 2^{\Delta+1} \) and the smallest \( \Delta \) satisfying the condition of part (iii) we have \( \langle d \rangle \leq 3.5n(n-1)(16n^2 + 11n \phi + 1) + 3.5(n+2)(n-1) + 14n(n-1) \), where \( \Delta = \phi + 5n \phi + 4n^2 \phi + 1 \).

(v) Let \( (h, h_0) \) belong to a linear description of \( P \), \( \|h\|_\infty > 0 \) and \( (h) + (h_0) \leq \phi \). Show that \( \langle \tilde{h} \rangle + \langle \tilde{h_0} \rangle \leq n \phi + 2 \) where \( \tilde{h} = h/\|h\|_\infty \) and \( \tilde{h_0} = h_0/\|h\|_\infty \).

(i) If \( x \in P_u \) is an extreme point of \( P_u \), then the assertion follows from point 7.5(b). So suppose that \( x \in P_u \) is an extreme point of \( P_u \) that is not an extreme point of \( P \). Then by point 7.2(b) \( x \) is determined uniquely by a system of equations

\[
\begin{pmatrix}
I_k & O \\
F_1 & F_2
\end{pmatrix}
\begin{pmatrix}
x^1 \\
x^2
\end{pmatrix}
= \begin{pmatrix} u g^* \\
f^*
\end{pmatrix}
\]

where \( I_k \) is the \( k \times k \) identity matrix with \( 1 \leq k \leq n \), \( F_1, F_2 \) are \( (n-k) \times k \) and \( (n-k) \times (n-k) \) matrices. \( g^* \) is a vector with entries +1 if \( x_j = u \), -1 if \( x_j = -u \) and \( f^* \) has \( n-k \) components.
Moreover, every row \((f^i, f_i)\) of \(\left( F_1 \ F_2 \ F^* \right)\) satisfies \(\langle f^i \rangle + \langle f_i \rangle \leq \phi\) and if \(k < n\) then \(\operatorname{det} F_2 \neq 0\). If \(k = n\) then \(F_2\) is empty and we define \(\operatorname{det} F_2 = 1\). Denote by \(G\) the \(n \times n\) matrix of this equation system and suppose the components of \(x\) are indexed to agree with the above partitioning into \(x^1\) and \(x^2\). By \(g_j\) we denote the \(j\)-th column of \(\left( I_k \ O \right)\), by \(f_j\) the \(j\)-th column of \(\left( F_1 \ F_2 \right)\) and by \(u_j\) the \(j\)-th unit vector in \(\mathbb{R}^n\). If we let

\[
G_j = G + u_j^T \begin{pmatrix} u g^* \\ f^* \end{pmatrix} - \begin{pmatrix} g_j \\ f_j \end{pmatrix},
\]

then by Cramer’s rule \(x_j = \det G_j / \det G\) and we need to estimate the digital sizes of the determinants. From formula (7.18) we get \(\langle \det G \rangle \leq 2(G) - n^2 \leq 2n\phi - n^2\). Moreover, \(\det G\) is a rational number of digital size less than \(2n\phi\) and thus there exist integers \(p, q\) with \(0 \leq |p| < 2^{2n\phi}\), \(1 \leq q < 2^{2n\phi}\) such that \(\det G = p/q\). Suppose that \(1 \leq j \leq k\). We calculate \(\det G_j = \pm u \det F_2\). Moreover, by the same reasoning as before \(\langle \det F_2 \rangle < 2n\phi\) is correct and thus there exist integers \(p_j, q_j\) with \(0 \leq |p_j| < 2^{2n\phi}\), \(1 \leq q_j < 2^{2n\phi}\) such that \(\det F_2 = p_j/q_j\). It follows that \(x_j = \pm u p_j q_i / q_j \) satisfies \(u p_j q_i < 2^{2n\phi} / q_j \) and \(1 \leq q_j \) such that \(u \ det F_2 = p_j / q_j\). We calculate also \(f_s \ det F_2 = \det (F_2 + v_j^T (f^* - f_j))\) by factoring out \(F_2\) and thus by (7.18) \(f_s \ det F_2 < 2n\phi\) is correct, i.e., there exist integer \(p_j, q_j\) with \(0 \leq |p_j| < 2^{2n\phi}\), \(1 \leq q_j < 2^{2n\phi}\) such that \(f_s \ det F_2 = p_j / q_j\). We calculate also

\[
-g_s \ det F_2 = \det \left( \begin{pmatrix} I_k & O \\ F_1 & F_2 \end{pmatrix} + u_j^T \begin{pmatrix} g^* \\ -f_j \end{pmatrix} \right)
\]

by applying the determinant formula for partitioned matrices first and then factoring out \(F_2\). Since \(\phi \geq n + 1\) it follows that the matrix on the right is a rational number of digital size less than \(2n\phi\) and thus there exist integers \(r, s\) with \(0 \leq |r| < 2^{2n\phi}\) and \(1 \leq s < 2^{2n\phi}\). Consequently, since \(u \geq 1\) is integer \(x_j = q (sp_j + u + q_j) / psq_j\) satisfies the assertion and the proof of (i) is complete.

(ii) By part (i) of this exercise we have \(x_j = p_j / q_j\) and \(y_j = r_j / s_j\) with integer numbers \(p_j, q_j, r_j, s_j\) satisfying \(0 \leq |p_j|, |r_j| < 2^{6n\phi+1}\) and \(1 \leq q_j, s_j < 2^{6n\phi}\) for \(1 \leq j \leq n\). Let \(c_j = a_j / b_j\) with integer \(a_j\) and \(b_j\) satisfying \(0 \leq a_j, b_j < 2^{6n\phi+1}\) for \(1 \leq j \leq n\). Let \(c \geq \) cy then it follows that \(c(x - y) \geq (\prod_{j=1}^n q_jsjb_j)^{-1} > 2^{-2n^2\phi-(c)}\) because \(\prod_{j=1}^n b_j < 2^c\).

(iii) The proof of this part of the exercise goes like the proof of Exercise 7.14(iii).
(iv) The proof follows from Exercise 7.14(iv) and by a simple substitution of the values for \( \Delta, \Lambda, u \) and the rough estimation \( 1 + 2^{3+\alpha} < 2^{4+\alpha} \) for \( \alpha \geq 0 \).

(v) Let \( h_j = \frac{p_j}{q_j} \) with integer \( p_j, q_j \) satisfying \( 0 \leq |p_j| < 2^\phi \), \( 1 \leq q_j < 2^\phi \) and \( |\frac{p_j}{q_j}|^2 \geq |\frac{p_i}{q_i}|^2 \) for \( 1 \leq j \leq n \). Then \( \tilde{h}_j = \frac{h_{j/i}}{\sqrt{\gamma_j}} \). Let \( \tilde{h}_i = 1 \). \( (\tilde{h}_j) \leq (p_j) + (q_j) + (p_i) + (q_i) \) and \( (\tilde{h}_i) = 2 \) where \( 0 \leq j \neq i \leq n \). Consequently, \( (\tilde{h}) + (\tilde{h}_0) \leq \langle h \rangle + (n - 1)(p_0 + q_0) + 2 \leq n\phi + 2 \).

*Exercise 9.11

(i) Let \( P = \{ x \in \mathbb{R}^2 : -4x_1 + 2x_2 \leq 1, x_1 + x_2 \geq 2, -x_1 + 2x_2 \leq 4, 2x_2 \geq 1 \} \) and \( P_u \) be as defined in (9.76). Find the maximizer \( x_{\text{max}} \) of \( \max \{ x_1 + x_2 : x \in P_u \} \) for \( u = 3 \) and the corresponding \( y_{\text{max}} \) for \( v = 2u \). Does \( y_{\text{max}} - x_{\text{max}} \) belong to the asymptotic cone of \( P \)? If not, what is the smallest possible value of \( u \) that works? What is the theoretical value that you get for \( u \) using \( \Lambda = \phi + 5n\phi + 4n^2\phi + 1 \)?

(ii) Suppose that the direction vector \( t \in P_u \) of the proof of Remark 9.19 satisfies \( ct = 0 \) and let \( T_u = \{ y \in T : cy = 0, \| y \|_\infty = 1 \} \) where (\( S, T \)) is a minimal generator of the polyhedron \( P \subseteq \mathbb{R}^n \) such that \( (x_j) \leq 4n\phi \) for all \( j \) and \( x \in S \cup T \). Prove that \( t \geq y \) for all \( y \in T_u \), i.e. that \( t \) is lexicographically greater than or equal to every \( y \in T_u \).

(iii) Determine the facet and vertex complexity of the polytopes \( S_n \) and \( C_n \) of Exercise 7.2 and of \( H_n \) and \( O_n \) of Exercise 7.7.

(iv) Find polynomial-time algorithms that solve the polyhedral separation problem over \( S_n, C_n, H_n \) and \( O_n \).

(i) The polytopes \( P_3 \) and \( P_6 \) are shown in the figure. Maximizing the function \( x_1 + x_2 \) over \( P_3 \) we get \( x_{\text{max}} = (3, 3) \) while maximizing it over \( P_6 \) we get \( y_{\text{max}} = (6, 5) \). The difference vector \( y_{\text{max}} - x_{\text{max}} = (3, 2) \) and it is not in the asymptotic cone \( C_\infty \) of \( P \), where \( C_\infty = \{ y \in \mathbb{R}^2 : -4x_1 + 2x_2 \leq 0, x_1 + x_2 \geq 0, -x_1 + 2x_2 \leq 0, 2x_2 \geq 0 \} \) since it violates the third inequality. Selecting \( u \geq 4 \) we have the maximizer lying on the extreme ray \( -x_1 + 2x_2 = 4 \) of the polyhedron \( P \) and thus \( u = 4 \) and \( v = 2u = 8 \) will give \( x_{\text{max}} = (4, 4) \) and \( y_{\text{max}} = (8, 6) \), and thus \( y_{\text{max}} - x_{\text{max}} = (4, 2) \in C_\infty \). To compute the theoretical value of \( u \), we first calculate \( \phi \) for the polyhedron \( P \). Since \( 0 = 1, (1) = (-1) = 2, (2) = 3 \) and \( (4) = (-4) = 1 \), \( (1) + (1) + (2), (-1) + (2) + (4), (0) + (2) + (1) \) we get that \( \phi \geq \max \{ (-4) + (2), (1), (1) + (1) + (2), (-1) + (2) + (4), (0) + (2) + (1) \} = 9 \). So selecting \( \phi = 9 \) we get from \( \Lambda = \phi + 5n\phi + 4n^2\phi + 1 \) with \( n = 2 \) that \( \Lambda = 244 \) and thus \( u = 2^{244} \) which is a horribly big number and evidently much bigger than required in this case.

(ii) By assumption the unique maximizer \( x_{\text{max}} \) of \( dx \) over the polytope \( P_u \) for \( u = 2^\Lambda \) satisfies \( x_{j\text{max}} = \pm u \) for at least one \( j \in \{1, \ldots, n\} \) and thus \( x_{\text{max}} \) is the unique solution to a system of
The polyhedron defining the proof of Exercise 9.10(i).) From the uniqueness of the respective maximizers \( x \) where \( \pm \) and \(-\) does not work in either case since it may, in the worst case, require exponential time to check them one by one for violation.) Since both \( S_n \) and \( O_n \) have exponentially many inequalities, \(-\) and \(-\) does not work in either case since it may, in the worst case, require exponential time.

\[
\begin{pmatrix}
\pm I_k & 0 \\
F_1 & F_2
\end{pmatrix}
\begin{pmatrix}
x_1^{\text{max}} \\
x_2^{\text{max}}
\end{pmatrix} =
\begin{pmatrix}
ue_k \\
f^*
\end{pmatrix},
\]

(1)

where \( \pm I_k \) is some \( k \times k \) matrix with \( 1 \leq k \leq n \) having \( \pm 1 \) or \(-1\) on its main diagonal (according to \( x_j = u \) or \( x_j = -u \), zeros elsewhere, \( F_1, F_2 \) are \((n-k) \times k\) and \((n-k) \times (n-k)\) matrices, \( e_k \) is a vector of \( k \) ones and \( f^* \) has \( n-k \) components. Every row \((f^*, f_i)\) of \( \begin{pmatrix} F & F_2 & f^* \end{pmatrix} \) satisfies \( (f^*) + (f_i) \leq \phi \) and \( \det F_2 \neq 0 \), where by convention \( \det F_2 = 1 \) if \( F_2 \) is empty. Moreover, \( y^{\text{max}} \) satisfies (1) with \( u \) replaced by \( v = 2^{\Lambda+1} \) and \( t = 2^{-\Lambda}(y^{\text{max}} - x^{\text{max}}) \in C_\infty \) satisfies \( (t_j) \leq 4n\phi \) for \( 1 \leq j \leq n \). (For more detail than given in the proof of Remark 9.17 on the estimation of \( (t_j) \) see the proof of Exercise 9.10(i).) From the uniqueness of the respective maximizers \( x^{\text{max}}, y^{\text{max}} \) and the assumptions that \( x_j = \pm u \) for at least one index \( j \) and \( \Lambda > 4n\phi \), it follows that every basis defining \( x^{\text{max}} \) is of the form (1) and thus by the duality theory of linear programming

\[
d = \lambda \begin{pmatrix}
\pm I_k & 0 \\
F_1 & F_2
\end{pmatrix}
\]

with \( \lambda \geq 0 \)

(2)

for some such basis defining \( x^{\text{max}} \) and \( y^{\text{max}} \), respectively. Consequently, a basis satisfying (1) and (2) exists. Since by construction \( t \in C_\infty \) and \( t \neq 0 \), it follows that \( \tilde{t} = t/\|t\|_\infty \in P_\infty \) is an extreme point of \( P_\infty \). More precisely, by dropping some of the constraints of \( P_\infty \) we have

\[
P_\infty \subseteq OC(\tilde{t}, \tilde{H}) = \left\{ y \in \mathbb{R}^n : \begin{pmatrix}
\pm I_k & 0 \\
F_1 & F_2
\end{pmatrix} y \leq \begin{pmatrix}
e_k \\
0
\end{pmatrix} \right\},
\]

i.e., \( OC(\tilde{t}, \tilde{H}) \) is the displaced outer cone with apex at \( \tilde{t} \) containing all of \( P_\infty \) (see the end of Chapter 7.5.4), and \( \tilde{t} \) is an extreme point of \( OC(\tilde{t}, \tilde{H}) \). From (2) it follows that \( \tilde{t} \) maximizes \( dy \) over \( OC(\tilde{t}, \tilde{H}) \) and thus by the outer inclusion principle, \( \tilde{t} \) maximizes \( dy \) over \( P_\infty \). The matrix \( \tilde{H} \) defining \( P_\infty \) is given by \( H, I_n \) and \(-I_n \) and thus \( P_\infty \) has a facet complexity of \( \phi \) – just like the polyhedron \( P \). Hence \( (\tilde{t}_j) \leq 4n\phi \) for \( 1 \leq j \leq n \). Since \( dt \geq dy \) for all \( y \in P_\infty \), it follows like in Exercise 7.14(iii) that \( \tilde{t} \) is the unique maximizer of \( d \) over \( P_\infty \) because the number \( \Delta \) that we use to prove Remark 9.19 is greater than the number \( 1 + 2^{4n\phi + 8n^2\phi(\phi+1)} \) that suffices to guarantee uniqueness. But then by Exercise 7.14(v) \( \tilde{t} \geq y \) for all \( y \in T_\infty \) and the proof of part (ii) is complete.

(iii) The digital size of the inequality \(-x_j \leq 0 \) equals \( n+2 \) since we also store zero coefficients. Likewise the digital size of \( x_j \leq 1 \) is \( n+3 \) and of \( \sum_{j=1}^{n} x_j \leq 1 \) is \( 2(n+1) \). Consequently, the facet complexity of the polytope \( S_n \) is \( \phi = 2(n+1) \) and that of \( C_n \) is \( \phi = n+3 \), where \( n \geq 1 \) is arbitrary. The polyhedron \( H_n \) of Exercise 7.7(i) has the same facet complexity as \( S_n \), i.e., \( \phi = 2(n+1) \), while the polytope \( O_n \) of Exercise 7.7(ii) has \( \phi = 2n+1 + \lceil \log_2 n \rceil \).

(iv) Since \( S_n \) has \( n+1 \) constraints and \( C_n \) has \( 2n \) constraints, the (trivial) algorithm LIST-and-CHECK is a polynomial-time separation algorithm for \( S_n \) and \( C_n \), respectively, no matter what rational \( y \in \mathbb{R}^n \) is given as input. (LIST-and-CHECK is just that; you list all inequalities and check them one by one for violation.) Since both \( H_n \) and \( O_n \) have exponentially many inequalities, LIST-and-CHECK does not work in either case since it may, in the worst case, require exponential time.
in $n$ to execute it. In the case of $H_n$ every constraint is of the form $h x = \sum_{j=1}^n \delta_j x_j \leq 1$ where $\delta_j \in \{0, 1\}$ for $1 \leq j \leq n$ and $\|h\|_{\infty} = 1$, except for the trivial constraint $0 x \leq 1$ which is never violated. The polyhedral separation problem for $H_n$ is

$$\max \{ \sum_{j=1}^n \delta_j z_j - 1 : \delta_j \in \{0, 1\} \text{ for } 1 \leq j \leq n \}$$

where $z \in \mathbb{R}^n$ is a rational vector. To solve the problem we scan the vector $z$ and set $\delta_j = 1$ if $z_j > 0$, $\delta_j = 0$ otherwise. This separation algorithm is linear in $n$ and $\langle z \rangle$ and thus a polynomial-time algorithm for the polyhedral separation problem for $H_n$.

In the case of the polytope $O_n$ we can check the $2n$ constraints $0 \leq x_j \leq 1$ for $1 \leq j \leq n$ by the algorithm LIST-and-CHECK in polynomial time. So we can assume WLOG that the rational vector $z \in \mathbb{R}^n$ for which we want to solve the polyhedral separation problem satisfies $0 \leq z_j \leq 1$. The separation problem for the remaining exponentially many constraints of $O_n$ is

$$\max \{ \sum_{j \in N_1} z_j - \sum_{j \in N-N_1} z_j - |N_1| + 1 : N_1 \subseteq N, |N_1| \text{ even} \} = 1 - \min \{ \sum_{j \in N_1} (1 - z_j) + \sum_{j \in N-N_1} z_j : N_1 \subseteq N, |N_1| \text{ even} \},$$

and a violated constraint is obtained if the objective function of the minimization problem is less than one. To solve the problem we order the components of $z$ in decreasing order which requires time that is polynomial in $n$ and $\langle z \rangle$. E.g. the sorting algorithm HEAPSORT requires $O(n \log n)$ operations in the worst case. So we can assume WLOG that $1 \geq z_1 \geq z_2 \geq \cdots \geq z_k \geq 1/2 > z_{k+1} \geq \cdots \geq z_n \geq 0$, where $0 \leq k \leq n$. Finding the index $k$ or verifying that $k = 0$ can be done by scanning the ordered vector $z$ once, i.e., in time that is linear in $n$ and $\langle z \rangle$. If the index $k$ is even we set $N_1^* = \{1, \ldots, k\}$. If $k$ is odd we set $N_1^* = \{1, \ldots, k-1\}$ if $z_k + z_{k+1} < 1$, $N_1^* = \{1, \ldots, k+1\}$ otherwise. By construction $|N_1^*|$ is even, $z_i_1 + z_i_2 < 1$ for all $i_1 \neq i_2 \notin N_1^*$ and $z_i_1 + z_i_2 \geq 1$ for all $i_1, i_2 \in N_1^*$ in all cases. We claim that $S_1^*$ solves the minimization problem. Suppose not and let $S \subseteq N$ be an optimal solution. Then $|S|$ is even, $S \neq N_1^*$ and

$$z(S) = \sum_{j \in S} (1 - z_j) + \sum_{j \in N-S} z_j < \sum_{j \in N_1^*} (1 - z_j) + \sum_{j \in N-N_1^*} z_j = z(N_1^*).$$

If $|S| = |N_1^*|$ then by construction $\sum_{j \in S} z_j \leq \sum_{j \in N_1^*} z_j$ and $\sum_{j \in N-N_1^*} z_j \leq \sum_{j \in N-S} z_j$. But then $z(N_1^*) \leq z(S)$ and thus if $z(S) < z(N_1^*)$ then $|S| - |N_1^*|$ is an even number different from zero. Suppose first that $|S| \geq 2 + |N_1^*|$. Then there exists $i_1 \neq i_2 \in S$ such that $i_1, i_2 \notin N_1^*$. Let $S' = S - \{i_1, i_2\}$. We compute $z(S') = z(S) - 2(1 - z_{i_1} - z_{i_2}) \geq z(S)$ by the optimality of $S$ and thus $z_{i_1} + z_{i_2} \geq 1$ which is a contradiction because $i_1 \neq i_2 \notin N_1^*$. Suppose now that $|S| + 2 \leq |N_1^*|$. Then there exists $i_1 \neq i_2 \in N_1^*$ such that $i_1 \notin S, i_2 \notin S$. Let $S' = S \cup \{i_1, i_2\}$. We compute $z(S') = z(S) + 2(1 - z_{i_1} - z_{i_2}) \geq z(S)$ by the optimality of $S$ and thus $z_{i_1} + z_{i_2} \leq 1$. Since $i_1, i_2 \in N_1^*$ we get $z_{i_1} + z_{i_2} = 1$ and thus $z(S') = z(S)$. Consequently, $S'$ is optimal as well as $|S| < |S'| \leq |N_1^*|$, we can reapply the reasoning and after finitely many steps we arrive at a contradiction because the cardinality of an optimal $S$ is bounded by $|N_1^*|$ in this case. Consequently, the claim follows.

The polyhedral separation problem for $O_n$ can thus be solved in $O(\langle z \rangle n \log n)$ time for any rational $z \in \mathbb{R}^n$.  

9.8. EXERCISES
Exercise 9.12

(i) Consider the polytope $P = \{ x \in \mathbb{R}^2 : 2x_1 + x_2 \leq 2, x_1 \geq 0, x_2 \geq 0 \}$. Find minimal generators for the corresponding $SP$, $SP_{\infty}$ and $SP^*$ as defined in (9.80), (9.81) and (9.82). Show that every nonzero extreme point $(h, h_0)$ of $SP_{\infty}$ defines a facet $hx \leq h_0 + \varepsilon$ of the $\varepsilon$-solidification $P^1_{\varepsilon}$ of $P$ in the $\ell_1$-norm and vice versa, that every facet $hx \leq h_0 + \varepsilon$ with $\|h\|_{\infty} = 1$ of $P^1_{\varepsilon}$ defines an extreme point $(h, h_0)$ of $SP_{\infty}$ where $\varepsilon > 0$.

(ii) Do the same as in part (i) of this exercise for the polyhedron $P = \{ x \in \mathbb{R}^2 : 2x_1 - x_2 = 0, x_1 \geq 1 \}$.

(iii) Do the same as in part (i) of this exercise for the polyhedron $P = \{ x \in \mathbb{R}^2 : 2x_1 + x_2 \geq 5, x_1 - x_2 \geq -2, x_2 \geq 1 \}$. In addition, let $(f, f_0) = (-1, -1, -6)$ and solve the linear program \[
\max \{ -x_1 - x_2 + 6x_3 : (x_1, x_2, x_3) \in SP_{\infty}^* \}. \]
Does its optimal solution yield a most violated separator for $(f, f_0)$ and $SP$? If not, what is the most violated separator in this case?

(iv) Let $P \subseteq \mathbb{R}^n$ be any nonempty, line free polyhedron and $P^1_{\varepsilon}$ its $\varepsilon$-solidification with respect to the $\ell_1$-norm where $\varepsilon > 0$. Show that the extreme points of $SP_{\infty}$ as defined in (9.81) are in one-to-one correspondence with the facets of $P^1_{\varepsilon}$. What happens if $P$ is permitted to have lines?

(i) The polytope $P$, see Figure 9.13, has three extreme points $x^1 = (1, 0)$, $x^2 = (0, 2)$ and $x^3 = (0, 0)$. Consequently, the $h_0$-polar $SP$ of $P$ is given by $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 - h_0 \leq 0, 2h_2 - h_0 \leq$
where we have used that μ the polytope (7.5) and running the double description algorithm (or by hand calculation) we find that \( SP \) is pointed and has a minimal generator consisting of the eight extreme points
\[
(0, 0, 0), (-1, 0, 0), (0, -1, 0), (-1, -1, 0), (1, 1, 2), (1, -1, 1), (1, \frac{1}{2}, 1), (-1, 1, 2)
\]
and the extreme ray given by \((0, 0, 1)\). The set of normed separators \( SP_\infty^* \) for \( SP \) given by (9.82) is the polytope
\[
SP_\infty^* = \{ x \in \mathbb{R}^3 : 2x_1 + x_2 - 2x_3 \leq 0, -x_1 \leq 0, -x_2 \leq 0, -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1, 0 \leq x_3 \leq 1 \}.
\]
Using the double description algorithm (or by hand calculation) we find that a minimal generator has the following six extreme points
\[
(0, 0, 0), (0, 0, 1), (0, 1, \frac{1}{2}), (0, 1, 1), (1, 0, 1), (\frac{1}{2}, 1, 1).
\]
To answer the second part of this problem we first calculate the \( \epsilon \)-solidification \( P_\epsilon^1 \) in the \( \ell_1 \)-norm. To do so we proceed like in the proof of Exercise 9.7(vi). To calculate \( P_\epsilon^1 \) we thus have to project out the μ-variables from the polyhedron
\[
PP_\epsilon^1 = \{ (z, \mu) \in \mathbb{R}^4 : -z_1 - z_2 + \mu_1 + 2\mu_2 \leq \epsilon, -z_1 + z_2 + \mu_1 - 2\mu_2 \leq \epsilon, z_1 - z_2 - \mu_1 + 2\mu_2 \leq \epsilon, z_1 + z_2 - \mu_1 - 2\mu_2 \leq \epsilon, \mu_1 + \mu_2 \leq 1, \mu_1 \geq 0, \mu_2 \geq 0 \}\]
where we have used that \( x = 0 \) is an extreme point of \( P \). To do so we need a minimal generator of the cone (for the general definition see(7.8))
\[
C = \{ u \in \mathbb{R}^7 : u_1 - u_2 + u_3 - u_4 + u_5 - u_6 = 0, 2u_1 + 2u_2 - 2u_3 - 2u_4 + u_5 - u_7 = 0, u \geq 0 \}.
\]
Running the double description algorithm we find that \( C \) is pointed and has the following ten extreme rays
\[
(1, 0, 0, 1, 0, 0, 0), (0, 1, 1, 0, 0, 0, 0), (1, 0, 1, 0, 0, 2, 0), (0, 0, 1, 0, 2, 3, 0), (0, 0, 0, 0, 1, 1, 1),
(0, 0, 0, 1, 2, 1, 0), (0, 1, 0, 3, 4, 0, 0), (0, 1, 0, 0, 1, 0, 3), (1, 1, 0, 0, 0, 0, 4), (1, 0, 0, 0, 0, 1, 2).
\]
Consequently, we find (besides some trivial redundant inequalities) that \( P_\epsilon^1 \) is given by
\[
P_\epsilon^1 = \{ x \in \mathbb{R}^2 : 2x_1 + x_2 \leq 2 + 2\epsilon, x_1 + x_2 \leq 2 + \epsilon, -x_1 + x_2 \leq 2 + \epsilon, x_1 - x_2 \leq 1 + \epsilon, -x_1 - x_2 \leq \epsilon, -x_1 \leq \epsilon, -x_2 \leq \epsilon \}.
\]
From Figure 9.13, we see that every inequality of the linear description of \( P_\epsilon^1 \) corresponds to a nonzero extreme point of \( SP_\infty^* \) and vice versa. Note, however, that you have to normalize the first inequality of \( P_\epsilon^1 \) to get the correspondence. The extreme points of \( P_\epsilon^1 \) for \( \epsilon \geq 0 \) are
\[
(-\epsilon, 0), (0, -\epsilon), (1, -\epsilon), (1 + \epsilon, 0), (\epsilon, 2), (0, 2 + \epsilon), (-\epsilon, 2).
\]
(ii) The polyhedron \( P \), see Figure 9.14, is a flat consisting of the extreme point \( x = (1, 2) \) and the direction vector \( y = (1, 2) \). Consequently, the \( h_0 \)-polar \( SP \) of \( P \) is the cone \( SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 + 2h_2 - h_0 \leq 0, h_1 + 2h_2 \leq 0 \} \). Running the double description algorithm (or by hand calculation) we find that \( SP \) is a blunt cone, the basis of the lineality space of \( SP \) is given by \((-2, 1, 0)\) and the conical part of \( SP \) is generated by \((0, 0, 1), (-1, 0, -1)\). So a minimal generator of \( SP \) is given by \((-2, 1, 0), (2, -1, 0), (0, 0, 1), (-1, 0, -1)\). The set \( SP_\infty \) of normed separators for \( P \) is obtained by intersecting \( SP \) with the constraints \( -1 \leq h_j \leq 1 \) for \( j = 1, 2 \) as we are working with the \( \ell_1 \)-norm. Using the homogenization (7.5) and running the double description algorithm, we find that \( SP_\infty \) is a pointed polyhedron. Its minimal generator consists of the four extreme points \((-1, 1, 0), (1, -\frac{1}{2}, 0), (-1, -1, -3), \) and \((1, -1, -1)\), and the extreme ray given by \((0, 0, 1)\). The set of normed separators \( SP_\infty^* \) for \( SP \) is the polytope

\[
SP_\infty^* = \{ x \in \mathbb{R}^3 : 2x_1 - x_2 = 0, -x_1 + x_3 \leq 0, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1 \}.
\]

Using the double description algorithm we find that a minimal generator of \( SP_\infty^* \) has the following three extreme points \((0, 0, 0)\), \((\frac{1}{4}, 1, 0)\), and \((\frac{1}{2}, 1, \frac{1}{2})\). To answer the second part of this problem we calculate the \( \varepsilon \)-solidification \( P_\infty^1 \) of \( P \) in the \( \ell_1 \)-norm. To do so we proceed like in part (i). To calculate \( P_\infty^1 \) we thus have to project out variable \( \nu_1 \) from the polyhedron

\[
PP_\infty^1 = \{(z, \nu_1) \in \mathbb{R}^3 : -z_1 - z_2 + 3\nu_1 \leq -3 + \varepsilon, z_1 - z_2 + \nu_1 \leq -1 + \varepsilon, -z_1 + z_2 - 3\nu_1 \leq 1 + \varepsilon, z_1 + z_2 - 3\nu_1 \leq 3 + \varepsilon, \nu_1 \geq 0 \}
\]

where we have simply eliminated the \( \mu \) variable since it must equal one. We thus need a minimal generator of the cone

\[
C = \{ u \in \mathbb{R}^5 : 3u_1 + u_2 - u_3 - 3u_4 - u_5 = 0, u \geq 0 \}.
\]

Running the double description algorithm we get the following six extreme rays

\[
(1, 0, 0, 1, 0), (1, 0, 3, 0, 0), (0, 1, 0, 0, 1), (0, 3, 0, 1, 0), (0, 1, 1, 0, 0), (1, 0, 0, 0, 3).
\]
To answer the second part of this problem, we calculate the minimal generator of $\mu$ is the polytope $SP$ by intersecting $P$. To normalize the first and second inequalities of $P^J$ to get the correspondence. The three extreme points of $P^J$ for $\varepsilon \geq 0$ are $(1 - \varepsilon, 2), (1, 2 - \varepsilon), (1 + \varepsilon, 2)$. In addition we need the direction vector $y = (1, 2) of the extreme ray of $P$ for a minimal pointwise description of $P^J$.

(iii) The polyhedron $P$, see Figure 9.15, is an unbounded set having two extreme points $(1, 3)$ and $(2, 1)$, and two direction vectors $(1, 0)$, and $(1, 1)$ for its extreme rays. Consequently, the $h_0$-polar $SP$ of $P$ is the cone $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 - 3h_2 - h_0 \leq 0, 2h_1 - h_2 - h_0 \leq 0, h_1 \leq 0, h_1 + h_2 \leq 0\}$. Running the double description algorithm we find that $SP$ is a pointed cone having four extreme rays $(0, 0, 1), (0, -1, 3), (-2, 1, -5)$ and $(1, -1, -3)$. The set $SP_\infty$ of the normed separators for $P$ is obtained by intersecting $SP$ with the constraints $-1 \leq h_j \leq 1$ for $j = 1, 2$. Using the homogenization (7.5) and running the double description algorithm we find that $SP_\infty$ is a pointed polyhedron. Its minimal generator consists of the five extreme points

$$(0, 0, 0), (0, -1, 3), (-1, -1, 2), (-1, \frac{1}{2}, -\frac{1}{2}), (-1, 1, -3)$$

and the extreme ray given by the direction vector $(0, 0, 1)$. The set of normed separators $SP_\infty$ for $SP$ is the polytope $SP_\infty = \{x \in \mathbb{R}^3 : 2x_1 + x_2 - 5x_3 \geq 0, x_1 - x_2 + 2x_3 \geq 0, x_2 - x_3 \geq 0, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1\}$. Running the double description algorithm we find that the (quasi-unique) minimal generator of $SP_\infty$ has the six extreme points

$$(0, 0, 0), (1, 0, 0), (1, 1, 0), (1, \frac{1}{2}, \frac{1}{2}), (1, 1, \frac{3}{2}), (\frac{3}{2}, 1, \frac{1}{2})$$

To answer the second part of this problem, we calculate the $\varepsilon$-solidification $P^J$ of $P$ in the $\ell_1$-norm. To do so we proceed like in parts (i) and (ii). To calculate $P^J$ we thus have to project out the $\mu$ and $\nu$ variables from the polyhedron

$$(z, \mu, \nu) = \{(z, \mu, \nu) \in \mathbb{R}^6 : z_1 + z_2 - 4\mu_1 - 3\mu_2 - \nu_1 - 2\nu_2 \leq \varepsilon, -z_1 + z_2 - 2\mu_1 + \mu_2 + \nu_1 \leq \varepsilon, z_1 - z_2 + 2\mu_1 - \mu_2 - \nu_1 \leq \varepsilon, -z_1 - z_2 + 4\mu_1 + 3\mu_2 + \nu_1 + 2\nu_2 \mu_1 + \mu_2 = 1, \mu \geq 0, \nu \geq 0\}.$$

To carry out the projection we calculate the cone (7.8):

$$C = \{(u, v) \in \mathbb{R}^3 : -4u_1 - 2u_2 + 2u_3 + 4u_4 - u_5 + v_1 = 0, -3u_1 + u_2 - u_3 + 3u_4 - u_6 + v_1 = 0, -u_1 + u_2 - u_3 + u_4 - u_7 = 0, -2u_1 + 2u_4 - u_8 = 0, u \geq 0\}.$$

Running the double description algorithm we get the seven extreme rays

$$(1, 0, 0, 1, 0, 0, 0, 0, 0, 0), (0, 1, 1, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 1, 1, 0, 0, 1), (0, 1, 0, 0, 0, 3, 1, 0, 2), (0, 0, 0, 1, 0, 1, 2, -3), (0, 1, 0, 3, 0, 0, 4, 6, -10), (0, 0, 1, 1, 4, 0, 2, -2)$$

and thus we calculate

$$P^J = \{x \in \mathbb{R}^2 : 2x_1 + x_2 \geq 5 - 2\varepsilon, x_1 - x_2 \geq -2 - \varepsilon, x_2 \geq 1 - \varepsilon, x_1 + x_2 \geq 3 - \varepsilon\}.$$
Let \( SP \) be a most violated separator for \( P \). Then \( SP \) corresponds to a nonzero extreme point of \( SP_\infty \) and vice versa. The extreme points of \( P_\varepsilon \) are \((1 - \varepsilon, 3), (2 - \varepsilon, 1), (2, 1 - \varepsilon)\). In addition we need the two direction vectors of the extreme rays of \( P \) for a minimal pointwise description of \( P_\varepsilon \).

From the above pointwise description of \( SP_\infty \) we find that \( \tilde{x} = (1, 1, \frac{1}{2}) \) is the unique optimal solution to \( \max \{ f x : x \geq 0, x \in P_\varepsilon \} \) and since \( f_i x_1 + f_2 x_2 - f_3 x_3 = 1.6 > 0 \) the point \( \tilde{x} \) separates \((f, f_0)\) from the cone \( SP \). To find the most violated separator for \((f, f_0)\) and \( SP \), we apply the procedure described on pages 344-346; see (9.83) and (9.84). Solving \( \max \{ f x : x \in P \} \) we find the (unique) optimizer \( x^{\max} = (2, 1) \) and thus (9.83) applies. We get \( \alpha = 1/2 \) and thus \( a^0 = (1, 1, \frac{1}{2}) \) is a most violated separator for \((f, f_0)\) and \( SP \), i.e., a most violated separator for \((f, f_0)\) and the cone \( SP \) cannot be found by solving the linear program \( \max \{ f x - f_0 x_{n+1} : (x, x_{n+1}) \in SP_\infty \} \).

(iv) Let \((h, h_0)\) be an extreme point of \( SP_\infty \). Then \( h x \leq h_0 \) for all \( x \in P \) and by Exercise 9.7(iv) \( h x \leq h_0 = h_0 + \varepsilon \) for all \( x \in P_\varepsilon \) since \( \|h\|_\infty = 1 \). Since \( \varepsilon > 0 \) and by Exercise 9.7(ix) \( \dim P_\varepsilon = n \), there exists \((f, f_0) \in \mathbb{R}^{n+1} \) such that \( \|f\|_\infty = 1 \), \( f x \leq f_0 \) defines a facet of \( P_\varepsilon \) and \( A = \{x \in P_\varepsilon : f x = f_0\} \). By Exercise 9.7(iv) \( f x \leq f_0 = \tilde{f}_0 - \varepsilon \) for all \( x \in P \) and thus \((f, f_0) \in SP_\infty \). From \( A \subseteq B = \{x \in P_\varepsilon : f x = f_0\} \). By Exercise 9.7(iv) \( f x \leq f_0 = \tilde{f}_0 - \varepsilon \) for all \( x \in P \) and thus \((f, f_0) \in SP_\infty \). From \( A \subseteq B \) it follows that \( h x^i = h_0 \) implies \( f x^i = f_0 \) for \( 1 \leq i \leq p \) and likewise \( h y^j = h_0 \) implies \( f y^j = f_0 \) for \( 1 \leq j \leq r \). Suppose \( h_j = \pm 1 \) for some \( j \in \{1, \ldots, n\} \). Since \( P \) is pointed, \( h x^i = h_0 \) for some \( i \in \{1, \ldots, p\} \). But \( x^i = \pm u_j \in P_\varepsilon \), \( h(x^i = \pm u_j) = h_0 = h \tilde{x}_j = h_0 \) and thus \( f(x^i = \pm u_j) = f_0 = \varepsilon h_j = f_0 = f_0 + \varepsilon \) implies \( f_j = h_j \). Since \((h, h_0)\) is an extreme point it follows that \( h x^i = (f, f_0) \), i.e., \( h x \leq h_0 + \varepsilon \) defines a facet of \( P_\varepsilon \). To show the reverse statement, suppose \((h, h_0) \in \mathbb{R}^{n+1} \) defines a facet of \( P_\varepsilon \). We can assume WLOG that \( \|h\|_\infty = 1 \) and thus \((h, h_0) \in SP_\infty \) where \( h_0 = \tilde{h}_0 - \varepsilon \). Denote by \((h^i, h_0^i)\) for \( 1 \leq i \leq s \) the extreme points of \( SP_\infty \). Since \( SP_\infty \) has exactly one halfline, it follows that \((h, h_0) = \sum_{i=1}^s \mu_i(h^i, h_0^i) + \lambda(0,1) \) with \( \mu_i \geq 0 \), \( \sum_{i=1}^s \mu_i = 1 \) and \( \lambda \geq 0 \). Suppose \( \lambda > 0 \). Then \( h x \leq h_0 - \lambda \) for all \( x \in P_\varepsilon \), because \((h, h_0 - \lambda)\) is a nonnegative combination of \( h^i x \leq h_0^i - \lambda \) with which the first part define facets of \( P \). But then \( h x \leq h_0 - \lambda + \varepsilon \) if \( x \in P_\varepsilon \) shows the contradiction. Consequently, \( \lambda = 0 \) and thus \((h, h_0) = \sum_{i=1}^s \mu_i(h^i, h_0^i) \) with

\[\text{Fig. 9.15. } \varepsilon\text{-solidifications of } P \text{ for } \varepsilon = 0, 1/2 \text{ and } 1\]
\[ \mu_i \geq 0, \sum_{i=1}^{s} \mu_i = 1, \text{ where } \tilde{h}_i^0 = h_i^0 + \varepsilon \text{ for } 1 \leq i \leq s. \] Since \( \dim P_\varepsilon = n \) it follows that \((h, \tilde{h}_0) = (h^i, \tilde{h}_i^0)\) for some \( i \in \{1, \ldots, s\} \) since the linear description of a full dimensional polyhedron by its facets is unique modulo the multiplication by positive scalars; see page 129 of the book. Consequently, \((h, h_0)\) defines an extreme point of \( SP_\infty \) and the proof is complete.

As we did not utilize the extremality of \( x^1, \ldots, x^p \) in the above argument it follows that the statement about the correspondence remains correct if \( P \) contains lines. If \( p = 0 \), then the feasible \( x^i \) needed to prove that \( f_j = h_j \) can be chosen to equal 0 since 0 \( \in P \) in this case.
Linear Optimization and Extensions
Problems and Solutions
Alevras, D.; Padberg, M.W.
2001, IX, 449 p. 37 illus., Softcover
ISBN: 978-3-540-41744-6