Optimality conditions for linear programs

Thesis for the Degree of Doctor of Philosophy (PhD)

by Norbert Tóth

Supervisor: Dr. Mihály Bessenyei

University of Debrecen
Doctoral Council for Natural Sciences and Engineering
Doctoral School of Mathematical and Computational Sciences
Debrecen, 2025

Hereby I declare that I prepared this thesis within the Doctoral Council for Natural Sciences and Engineering, Doctoral School of Mathematical and Computational Sciences, University of Debrecen, in order to obtain a PhD Degree in Natural Sciences at the University of Debrecen.
The results published in the thesis are not reported in any other PhD theses.

Debrecen, 20……………………

……………………….
signature of the candidate

Hereby I confirm that Norbert Tóth, PhD candidate conducted his studies with my supervision within the Mathematical analysis, functional equations and inequalities Doctoral Program of the Doctoral School of Mathematical and Computational Sciences, University of Debrecen, between 2021 and 2025. The independent studies and research work of the candidate significantly contributed to the results published in the thesis. I also declare that the results published in the thesis are not reported in any other PhD theses.
I support the acceptance of the thesis.

Debrecen, 20……………………

…………………………
signature of the supervisor

Optimality conditions for linear programs

Dissertation submitted in partial fulfilment of the requirements
for the doctoral (PhD) degree in Mathematics and Computing

Written by Norbert Tóth
certified applied mathematician

Prepared in the framework of the Doctoral School of Mathematical
and Computational Sciences of the University of Debrecen
(Mathematical analysis, functional equations and inequalities doctoral programme)

Dissertation supervisor: Dr. Mihály Bessenyei

The official opponents of the dissertation:

Dr. ....
Dr. ....

The evaluation board:

Chairperson: Dr. ....
Members: Dr. ....
Dr. ....
Dr. ....
Dr. ....

The date and venue of the dissertation defence: ……………..………, 20……………

"Just do it!"

Acknowledgements

I wish to express my deep gratitude to everyone who was part of this adventurous journey: my Family, my Friends and my Colleagues in Debrecen and Miskolc. I would like to thank them for always supporting me, pushing me forward and giving encouragement when I lacked motivation.

I am deeply grateful to Éva Kaszanyi, with whom I could share or discuss my best and worst moments anytime over the past 9 years. Without her support I would not be where I am now.

I would also like to thank all the people in the Zion Team, especially to Nóra Benkovics for helping me start the journey of my personal development which greatly contributed to my PhD studies and my future plans. Furthermore, to Bessy Helmeczy for her powerful quote "Just do it!" which became my mantra and gave me the final push I needed to prepare the final version of my dissertation.

I owe my deepest gratitude to Mihály Bessenyei who saw the potential in me 7 years ago when I applied for a thesis topic during my BSc studies and became my supervisor. Since then, he has paved my way many times to get me this far, gave me excellent ideas and outstanding guidance.

Finally, I would like to thank all the other people who I could not list here by name, but contributed to the completion of my dissertation, let it be by providing me technical or emotional support.

Contents

Introduction

> *"For since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear." (Euler)*

Linear programs are constrained optimization problems in which the objectives and the feasible sets are given in terms of linear expressions. Their systematic study has begun in the first half of the 20th century. On the American side, Dantzig applied them to the transportation and storage problems of the US Air Force during World War II. His results were obtained in 1947, but they were not published until 1951 [**9**]. Independently and simultaneously on behalf of the Russians (Soviets), Kantorovich developed a similar method for optimizing the production of a furniture factory in 1939, and he also published it only many years later in [**16**]. Unfortunately, Kantorovich's work was not recognized in the West and it was unnoticed in the East in its time. In 1975, Kantorovich received the Nobel Prize in Economics for his fundamental work which demonstrates the importance of linear programming. However, Dantzig was not so lucky: The Royal Swedish Academy of Sciences considered his work "too mathematical".

The research of Kantorovich and Dantzig ultimately concluded that, in spite of their special form, linear programs are significant not only from a theoretical point of view and not only for the interested mathematical community, but from the direct real-world applications. The algorithm behind the solution, developed by Dantzig [**10**] and known as the *simplex method*, can be implemented easily via computer programs.

The aim of the present dissertation is to revisit and study two topics connected to linear programs. We present our results in four chapters in the following way, based on the materials of the papers [**5**], [**6**].

In CHAPTER 1, we recall the most important notions and tools of Convex Analysis and Convex Geometry that we will use throughout the dissertation. In several cases, we present alternative and elementary proofs of well-known results to make the discussion self-contained. Our basic references for this introductory chapter are the books by Barvinok [**1**], by Borwein and Lewis [**7**], and by Rockafellar [**24**].

In CHAPTER 2 we are going to study *log-barrier problems*, the modified versions of linear programs: The original objective function is replaced by a one-parameter family of functions, while the feasible set essentially remains unchanged. If the perturbed objectives have a unique optimum for each parameter, then the optimal solutions form a parametrized curve called the *central path*. As the parameter shrinks to zero, the perturbed objective approaches the original one. Thus the central path is expected to terminate in the optimal solution of the original linear program.

Although Vanderbei discusses the log-barrier problem in his excellent monograph [**29**], his reasoning is incorrect at a point: The Heine–Borel Theorem is applied in the relative topology induced by an open subset of $\mathbb{R}^n$. Our main result corrects this mistake and results in an extended version of the log-barrier problem. Moreover, it enlightens the distinguished role of the logarithmic perturbation, as well. In the proof, we use the methods of Convex Analysis combined with some ideas of Wright [**32**]. The advantages of our approach are that it completely avoids the second order optimality test, and it is independent of the simplex method.

Let us note here that interior point methods are closely connected to complexity questions of linear programs. As Klee and Minty [**19**] illustrate, the simplex method supplemented with Bland's Rule may terminate exponentially. Moreover, it turned out since then, that almost every known variant of the simplex method has the same feature. An unsolved problem is, whether there exists a version of the simplex method that runs in polynomial time.

In the geometrical point of view, the simplex algorithm moves on the boundary of the feasible set during pivoting. In the contrary, interior point methods – as their name suggests – approach the optimum on interior paths. In 1979, Khachiyan [**18**] developed the *ellipsoid method* and showed that it terminates in polynomial time. This proves that solving linear programs has polynomial complexity. The systematic study and the first easily implementable method is due to Karmarkar [**17**]. Later Megiddo [**21**] proved that Karmarkar's algorithm relying on projective geometry is actually equivalent to an interior point method by Fiacco and McCormick [**12**], which was named later as the *central path method* by Huard [**15**]. An excellent summary on the nonlinear geometry of the central path method is the survey by Bayer and Lagarias [**2, 3**]. For further theoretical details and implementation issues, we refer to Roos, Terlaky and Vial [**25**].

In CHAPTER 3, we are going to give a geometrical optimality condition for linear programs as a main result. The proof is based on Convex Geometry and suggests an algorithm which might differ from the simplex method. The main result is an extension of the *graphical method*, the well-known topic of introductory courses. For a more detailed discussion on the geometric properties of linear programming we recommend the book of Schrijver [**26**].

As applications, we give optimality conditions for problems in canonical and standard form, derive the strong duality theorem, and prove Farkas' lemma [**11**]. This lemma, the iconic representation of the theorems of alternatives, has many equivalents, like the results of Gordan [**14**], Motzkin [**23**], Stiemke [**27**], Tucker [**28**] and Ville [**30**]. For further proofs of Farkas' lemma, we recommend the books of Borwein and Lewis [**7**] or Gale [**13**]. An elegant exposition is due to Komornik [**20**].

In CHAPTER 4, we present an algorithm that decides the solvability of planar feasible linear programs. The theoretical background for the validation is the main result of the previous chapter. The Appendix contains the complete Maple code of the algorithm.

CHAPTER 1

Convex Analysis and Convex Geometry background

In this first chapter we revisit to the basic tools of Convex Analysis and Convex Geometry that we are going to use generally or in most of the latter chapters throughout the dissertation. One of the most important tool that we need to develop here is the recession cone of convex sets which will play crucial roles in the upcoming parts. Therefore, we dedicate a whole section to them to collect most of their necessary properties.

1.1. Auxiliary tools from Convex Analysis

First of all, let us recall the basic definition of (strictly) convex and concave functions.

DEFINITION. *Let X be a vector space, and let $D \subseteq X$ be a convex set. We say that $f\colon D \to \mathbb{R}$ is a* convex function *if, for all $\lambda \in [0,1]$ and for all $x, y \in D$,*

$$f\big(\lambda x + (1-\lambda)y\big) \leq \lambda f(x) + (1-\lambda)f(y).$$

If this inequality is strict for all $\lambda \in]0,1[$ and for all $x \neq y$, then we speak about a strictly convex function*. If the formula above holds with the opposite type (strict) inequality, then we say that f is a* (strictly) concave function*.*

As it can easily be checked, the set of convex (concave) functions on D forms a convex cone with respect to the pointwise operations. We quote here the result of Bernstein and Doetch [**4**]: *If a convex (concave) function acting on an open, convex subset of a normed space is locally bounded from above (below) at a point, then it is locally Lipschitz.* In the case of finite dimensional vector spaces this boundedness property holds automatically. Thus Lipschitz property (in particular: continuity) is fulfilled without any further restrictions: *Any convex (concave) function acting on an open, convex subset of a finite dimensional normed space is continuous.*

We also revisit here the basic notions for convex and cone combination of elements in a vector space, moreover, we define the convex and conic hulls of them, as well.

DEFINITION. *Let X be a vector space, $D \subseteq X$ a nonempty set and $k \in \mathbb{N}$ fixed. Then, the* convex combination *of the points $p_1, \dots, p_k \in D$ is $\lambda_1 p_1 + \cdots + \lambda_k p_k$ where the real numbers λ_i satisfy $\lambda_i \in [0,1]$ for all $i = 1, \dots, k$, and $\lambda_1 + \cdots + \lambda_k = 1$.*

DEFINITION. *Let X be a vector space and $k \in \mathbb{N}$ fixed. If $\{p_1, \dots, p_k\} \subseteq X$ is an affinely independent set, then* $\operatorname{conv}(p_1, \dots, p_k)$ *denotes its* convex hull*, that is,*

$$\operatorname{conv}(p_1, \dots, p_k) = \left\{ \sum_{i=1}^{k} \lambda_i p_i \;\middle|\; \lambda_1, \dots, \lambda_k \in [0,1], \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

The particular cases $k = 2$ and $k = 3$ are called a segment *and a* triangle*, respectively.*

DEFINITION. *Let X be a vector space, $D \subseteq X$ a nonempty set and $k \in \mathbb{N}$ fixed. Then, the* conic combination *of the points $p_1, \dots, p_k \in D$ is $\lambda_1 p_1 + \dots + \lambda_k p_k$ where the real numbers λ_i satisfy $\lambda_i \geq 0$ for all $i = 1, \dots, k$.*

DEFINITION. *Let X be a vector space and $D \subseteq X$ a nonempty set. Then, we call the set* $\operatorname{cone}(D)$ *the* conic hull *of D if it contains all the conic combinations of elements in D, that is,*

$$\operatorname{cone}(D) = \left\{ \sum_{i=1}^{k} \lambda_i p_i \;\middle|\; k \in \mathbb{N}, \forall i = 1, \dots, k \colon p_i \in D, \lambda_i \geq 0 \right\}.$$

The basic separation theorem plays a crucial role in optimization and several proofs are known for its Euclidean version. Barvinok argues with algebraic manipulations and isolation. Borwein and Lewis use Weierstrass' condition and the first order necessary condition. Barvinok also suggests an alternative approach as a posed problem. Now we elaborate its solution.

LEMMA 1.1. *In Euclidean spaces, any nonempty, closed and convex set can be strictly separated by a hyperplane from any point not belonging to the set.*

PROOF. Let C be a nonempty, closed and convex subset of a Euclidean space X and let p be an element of $X \setminus C$. Choose $x \in C$. Then $K := C \cap \overline{U}\big(p, d(p,x)\big)$ is closed and bounded, hence it is compact by the Heine–Borel Theorem. The continuity of the metric guarantees the existence of a nearest point $x_0 \in K$ to p. Since C is closed, $d(p, x_0) > 0$. Moreover, the distance of p from $C \setminus K$ is at least $d(p,x)$, while $d(p,x_0) \leq d(p,x)$. Therefore

$$d(p, x_0) = \min_{x \in C} d(p, x).$$

Consider now the hyperplane L that is normal to the segment $\operatorname{conv}(p, x_0)$ and contains its midpoint. We claim that L is a suitable hyperplane we are looking for. Indeed, assume to the contrary that L does not separate C and p; then there exists $y \in C$ such that $\operatorname{conv}(x_0, y)$ intersects L at a point y_0. In the triangle $\operatorname{conv}(p, x_0, y_0)$, the angle at the vertex x_0 is less then $\pi/2$. Thus we can choose a point $x \in \operatorname{conv}(x_0, y_0)$ such that the angle at the vertex x is the greatest in the triangle $\operatorname{conv}(p, x_0, x)$. Then $d(p,x) < d(p,x_0)$ and x belongs to C by convexity. This contradicts to the minimal distance property of x_0. Our argument works only if p, x_0, y_0 are not collinear. However, the collinear case can easily be excluded, as well. $\square$

From now on, let us denote the vector space of all linear maps from $\mathbb{R}^n$ to $\mathbb{R}^m$ by $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, and the nonnegative orthant of $\mathbb{R}^n$ by $\mathbb{R}^n_+$, that is, the set of vectors $x \in \mathbb{R}^n$ satisfying $x_i \geq 0$ for all $i \in \{1, \ldots, n\}$. Considering linear transformations, they may destroy the closedness of (convex) sets even in very simple cases. However, if the set is a polyhedron, then its linear transform still remains a polyhedron. The proof of this fact is surprisingly complicated; for those who are interested in the details we refer to Barvinok's solution. Fortunately, we can avoid inconvenience with the next observation that will be sufficient in our reasoning. In the proof, we use the conical Carathéodory theorem [**8**].

LEMMA 1.2. *Any finitely generated cone of a vector space is the union of the independently generated subcones. In particular, any finitely generated cone of a finite dimensional space is closed.*

PROOF. Denote the family of independently generated subcones of the original cone C by $\mathcal{C}$. Clearly $\bigcup \mathcal{C} \subseteq C$. Assume now that $x \in C$. Since C is finitely generated, it is a subset of a d-dimensional space. By the Carathéodory theorem, there exists an at most d element subset F of the original generator such that $x \in \operatorname{cone}(F)$. If F is independent, then we are done by the first observation. If not, then we can repeat the previous observation and can guarantee an at most $(d-1)$ element subset of F such that $x \in \operatorname{cone}(F)$. This process finally terminates since we can reduce the dimension (and the cardinality of the generator) at each step. In the final step, we obtain a linearly independent subsystem of the original one whose cone hull contains x. That is, $C \subseteq \bigcup \mathcal{C}$.

For the second statement, observe that independently generated cones in a finite dimensional space are closed by the Bolzano–Weierstrass theorem. On the other hand, finitely generated cones are the finite union of independently generated closed subcones. However, the finite union of closed sets are closed, as well. $\square$

The book of Borwein and Lewis suggests a "dual proof" of this statement. The key idea of this approach is that a finitely generated cone is obtained as the intersection of finite closed half-planes and hence it is closed. The main tool is the basic separation property of Lemma 1.1.

Let us define now constrained optimization problems in general form and some other notions connected to them which give the basic background for linear programming problems.

DEFINITION. *Let X and Y be nonempty sets, and let $D \subseteq X$ further $b \in Y$. Consider the constrained optimization problem*

$$\begin{aligned} f(x) &\longmapsto \max \\ g(x) &= b \end{aligned}$$

where the objective function $f\colon D \to \mathbb{R}$ *and the* constraint $g\colon X \to Y$ *are given. We say that* $x \in X$ *is a* feasible solution *if* $x \in D$ *and* $g(x) = b$*. Furthermore, the set of feasible solutions is called the* feasible set *of the problem. An element* $x^* \in X$ *is called a feasible* optimal solution*, if it is feasible and* $f(x) \leq f(x^*)$ *whenever* $x \in X$ *is a feasible solution.*

In particular, let $X = \mathbb{R}^n$ and $Y = \mathbb{R}^m$. We call the constrained optimization problem

$$\begin{gathered}\langle c, x\rangle \longmapsto \max\\ x \in P\end{gathered}$$

a *linear programming problem* where $c \in \mathbb{R}^n$ is a given vector and $P \subseteq \mathbb{R}^n$ is a *polyhedron*, that is, the intersection of finitely many half-spaces. The other notions can be analogously derived from the previous definition.

Finally, at some points we are going to work with linear programs in matrix form where inequalities will occur. The inequalities in these problems will refer to the *coordinatewise ordering*, that is, $u \leq v$ if and only if all coordinates of u is smaller than or equal to the corresponding coordinates of v. In notation we will not distinguish this ordering from the usual one since it will not cause any misunderstanding. For the same reason, we will use the symbol 0 to vectors and to the real number in most cases.

1.2. The recession cone of convex sets

As a first step in this section let us define the recessional directions of nonempty sets of a vector space and then discuss some of their properties, as well.

DEFINITION. *Let D be nonempty subset of a vector space X, and let $x \in D$ be fixed. If $x + \alpha r \in D$ for all $\alpha \geq 0$, then we say that $r \in X$ is a* recessional direction *of D at the point x.*

Rockafellar investigated the case when D is a *closed*, convex set, and it turns out that the recessional directions of D have many important properties. They are independent on the choice of $x \in D$ and the existence of recessional directions characterize the unboundedness of D. We formulate this statement in the next lemma, however, we omit its proof here; for those who are interested in the details we refer to Rockafellar's solution.

LEMMA 1.3. *If D is a nonempty, closed, convex subset of a Euclidean space X, and $r \in X$, then the next statements are equivalent:*

(1) There exists $x \in D$, such that $x + \alpha r \in D$ for all $\alpha \geq 0$;
(2) For all $x \in D$ and for all $\alpha \geq 0$, we have that $x + \alpha r \in D$.

Moreover, D is bounded if and only if the set of recessional directions of D is trivial.

Consequently, we can denote the set of recessional directions of D simply by $\operatorname{rec}(D)$. Furthermore, with some easy calculations it can be shown that the recessional directions form a cone: $\operatorname{rec}(D)$ is closed under addition and multiplication by nonnegative scalars. Thus, we will call $\operatorname{rec}(D)$ the *recession cone* of D.

It turns out that the same statements remain true if D is an *open*, convex set as the next lemma shows. Despite the similar properties, we have to follow an alternative approach in the proof. Due to the independence on the base point, we can still denote the recession cone of D by $\operatorname{rec}(D)$. We note here that using open sets instead of closed ones will be essential later: The objective's domain of the log-barrier problem requires it.

LEMMA 1.4. *If D is a nonempty, open, convex subset of a Euclidean space X, and $r \in X$, then the next statements are equivalent:*

1. *There exists $x \in D$, such that $x + \alpha r \in D$ for all $\alpha \geq 0$;*
2. *For all $x \in D$ and for all $\alpha \geq 0$, we have that $x + \alpha r \in D$.*

Moreover, D is bounded if and only if $\operatorname{rec}(D)$ *is trivial.*

PROOF. Clearly, it suffices to prove the implication $(1) \Rightarrow (2)$. Assume that $x + \beta r \in D$ holds for some $x \in D$ and for all $\beta > 0$. Let $x_0 \in D$ be arbitrary (see Figure 1.1). We may assume that x_0 does not lie on the line whose direction is r and passes trough x. By openness, x_0 belongs to D with some neighborhood U. Let $y \in U$ such that $x_0 \in [x, y]$. Consider a point $w := x_0 + \alpha r$, where $\alpha \geq 0$. Then the parallel lines (with direction r) determined by x and x_0, furthermore the point y are on the same plane. By the Axiom of Parallels, y and w determine a line which intersects the half-line $\{x + \beta r \mid \beta > 0\}$ at a point z. Then, $z \in D$ by assumption (1), and $w \in [y, z]$. That is, w can be represented as the convex combination of two elements of D. Since D is convex, $w \in D$ follows.

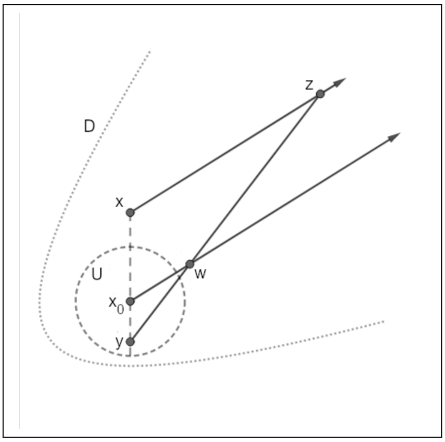


FIGURE 1.1

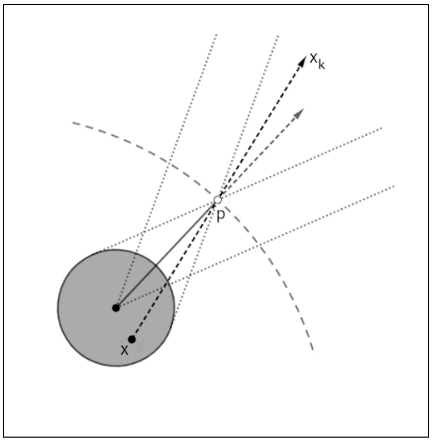


FIGURE 1.2

The necessity of the second statement is clear. Conversely, assume to the contrary that $\mathrm{rec}(D)$ is trivial while D is unbounded. Then, D contains an unbounded sequence $(x_k)_{k\in\mathbb{N}}$ (see Figure 1.2). By (2) and by the openness of D, we may assume that the closed unit ball with center at the origin is contained in D. Similarly, we may also assume that $(x_k)_{k\in\mathbb{N}}$ does not contain the zero vector. Define the sequence $(r_k)_{k\in\mathbb{N}}$ by

$$r_k := \frac{x_k}{\|x_k\|}.$$

This sequence is bounded since its members are unit vectors. The Bolzano–Weierstrass Theorem guarantees that $(r_k)_{k\in\mathbb{N}}$ has a limit point r; moreover, we may assume that $r_k \to r$ as $k \to +\infty$. By the indirect assumption, $r \notin \mathrm{rec}(D)$. Thus,

$$\alpha_0 := \sup\{\alpha \geq 0 \mid \alpha r \in D\} < +\infty.$$

Furthermore, the convexity of D ensures that $\alpha r \in D$ if $\alpha < \alpha_0$ and $\alpha r \notin D$ if $\alpha > \alpha_0$. In particular, these properties show that $p := \alpha_0 r \in \partial D$. Therefore, $p \notin D$ since D is open. Consider now the set defined by

$$C := \{\alpha(p - x) \mid \alpha \geq 0,\ \|x\| = 1\}.$$

The convexity of closed unit ball provides that C is a convex cone. Clearly, $r \in C$, and hence $r_k \in C$ for sufficiently large indices. Thus, $x_k \in p + C$ if k is large enough. However, in such cases, p can be obtained as a convex combination of x_k and a unit vector. Both of these elements belong to D, therefore $p \in D$ by convexity. This contradiction completes the proof. □

Finally, we also need to focus on the following calculus property of recession cones that we formulate for both the open and closed cases. For the sake of completeness, we also sketch the proof in both cases despite their similarities.

LEMMA 1.5. *If A and B are intersecting open, convex subsets of a Euclidean space X, then*

$$\mathrm{rec}(A \cap B) = \mathrm{rec}(A) \cap \mathrm{rec}(B).$$

PROOF. Note that $A \cap B$ is an open convex subset of both A and B. Thus $\mathrm{rec}(A\cap B)$ is a subset of the cones $\mathrm{rec}(A)$ and $\mathrm{rec}(B)$. For the converse inclusion, let $r \in \mathrm{rec}(A) \cap \mathrm{rec}(B)$ and let $x \in A \cap B$. Then $x + \alpha r \in A \cap B$ for all $\alpha \geq 0$. Thus $r \in \mathrm{rec}(A \cap B)$. □

LEMMA 1.6. *If A and B are intersecting closed, convex subsets of a Euclidean space X, then*

$$\mathrm{rec}(A \cap B) = \mathrm{rec}(A) \cap \mathrm{rec}(B).$$

PROOF. Note that $A \cap B$ is a closed convex subset of both A and B. Thus $\mathrm{rec}(A\cap B)$ is a subset of the cones $\mathrm{rec}(A)$ and $\mathrm{rec}(B)$. For the converse inclusion, let $r \in \mathrm{rec}(A) \cap \mathrm{rec}(B)$ and let $x \in A \cap B$. Then $x + \alpha r \in A \cap B$ for all $\alpha \geq 0$. Thus $r \in \mathrm{rec}(A \cap B)$. □

CHAPTER 2

A convex analysis view of the barrier problem

In this chapter we are going to study the *log-barrier problem* that we can define in a compact way as

$$\begin{aligned} c^T x + t \sum \log x &\longmapsto \max \\ Ax &= b \\ x &> 0 \end{aligned}$$

with parameter $t > 0$. A detailed approach is presented by Vanderbei which heavily depends on the simplex method. Moreover, the reasoning is incorrect at a point: The Heine–Borel Theorem is applied in the relative topology induced by an open subset of $\mathbb{R}^n$.

The main aim of this chapter is to generalize the perturbation in the objective of the barrier problem and correct Vanderbei's proof but with an alternative approach using methods from Convex Analysis instead of relying heavily on linear programming. The most important tool we use is the recession cone of *open*, convex sets. The openness assumption here will be clearly necessary since the classical barrier problem has the logarithm in its objective. At the end of the chapter we revisit the classical log-barrier problem and we present it as a direct application of our main results.

2.1. Auxiliary tools

Throughout this chapter we will work with the particular form of linear programs, as follows. Let $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ be given. Then the pair of linear programming problems

$$\text{Primal:} \quad \begin{aligned} c^T x &\longmapsto \max \\ Ax &\le b \\ x &\ge 0; \end{aligned} \qquad\qquad \text{Dual:} \quad \begin{aligned} y^T b &\longmapsto \min \\ A^T y &\ge c \\ y &\ge 0 \end{aligned}$$

is termed the *primal–dual pair in standard form.* The notion of feasible (optimal) solution remains the same as it was mentioned in the previous chapter. To achieve equality form in the constraints, let us introduce the nonnegative *slack variables* $\omega \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ in both problems respectively:

$$\begin{aligned} c^T x &\longmapsto \max & \qquad y^T b &\longmapsto \min \\ Ax + \omega &= b & A^T y - z &= c \\ x, \omega &\geq 0; & y, z &\geq 0. \end{aligned}$$

In linear programming it is well-known that the optimality of feasible primal–dual solution pairs can easily be checked by the Complementary Slackness Theorem:

THEOREM 2.1. *Let* $x = (x_1, \ldots, x_n)^T$ *and* $y = (y_1, \ldots, y_m)^T$ *be feasible solutions of the primal and the dual problems, and let* $\omega = (\omega_1, \ldots, \omega_m)^T$ *and* $z = (z_1, \ldots, z_n)^T$ *be the attached slack variables. Then* x *and* y *are optimal for their respective problems if and only if*

$$x_1 z_1 = \cdots = x_n z_n = 0 = \omega_1 y_1 = \cdots = \omega_m y_m.$$

We are going to study the central path in a geometrical point of view. To be able to do this, we are going to introduce first the path and its generator in an abstract setting and show their properties in two lemmas.

DEFINITION. *Let* T *and* B *be nonempty sets. We say that* $F\colon T \times B \to \mathbb{R}$ *fulfills the* parametrized global maximum property *if, for all* $t \in T$ *there exists a unique element* $h(t) \in B$ *such that*

$$F\big(t, h(t)\big) > F(t, x)$$

holds for all $x \neq h(t)$. *In this case,* $h\colon T \to B$ *is a function, indeed. The function* h *and the map* F *are called the* path *and its* generator, *respectively.*

Now assume that T is a metric space and B is a nonempty subset of a Euclidean space E, and consider the next limit conditions for $F\colon T \times B \to \mathbb{R}$: For all $t_0 \in T$ and for all $x_0 \in \overline{B} \setminus B$,

$$\lim_{\substack{t \to t_0 \\ x \to x_0}} F(t, x) = -\infty. \tag{2.1}$$

For all $t_0 \in T$,

$$\lim_{\substack{t \to t_0 \\ \|x\| \to +\infty}} F(t, x) = -\infty. \tag{2.2}$$

These requirements on the path generator ensure that the induced central path is continuous:

LEMMA 2.2. *Let* T *be a metric space, and* B *be a nonempty subset of a Euclidean space* E. *If* $F\colon T \times B \to \mathbb{R}$ *is a continuous path generator which has the limit conditions* (2.1) *and* (2.2), *then the path* $h\colon T \to B$ *is continuous.*

PROOF. Assume to the contrary that there exist $t_0 \in T$ and a sequence (t_k) in T such that $t_k \to t_0$ whereas $h(t_k) \nrightarrow h(t_0)$. Firstly, consider the case when $h(t_k)$ is bounded. Without the loss of generality we may assume that $h(t_k) \to x_0 \in \overline{B}$

by the Bolzano–Weierstrass Theorem. If $x_0 \in B$, then the continuity and the maximum property of F imply

$$\begin{aligned} F(t_0, x_0) &= \lim_{k\to\infty} F\big(t_k, h(t_k)\big) \\ &\geq \lim_{k\to\infty} F\big(t_k, h(t_0)\big) \\ &= F\big(t_0, h(t_0)\big) \\ &> F(t_0, x_0), \end{aligned}$$

which is a contradiction. If $x_0 \notin B$, then the properties of F and the limit condition (2.1) result in the contradiction

$$F\big(t_0, h(t_0)\big) = \lim_{k\to\infty} F\big(t_k, h(t_0)\big) \leq \lim_{k\to\infty} F\big(t_k, h(t_k)\big) = -\infty.$$

To complete the proof, we have to discuss the case when $h(t_k)$ is unbounded. This can be done similarly to the previous case, using the limit condition (2.2). □

For special generators, we have monotonicity along the central path:

LEMMA 2.3. *Let $I \subseteq \mathbb{R}$ be an interval, B be a nonempty subset of a Euclidean space E, $c \in E$, and $g\colon E \to \mathbb{R}$. If the function $F\colon I \times B \to \mathbb{R}$ defined by*

$$F(t, x) := c^T x + tg(x)$$

satisfies the parametrized global maximum property, then $g \circ h$ is monotone increasing.

PROOF. Let $t, s \in I$. By the parametrized global maximum property,

$$\begin{aligned} c^T h(t) + tg\big(h(t)\big) &\geq c^T h(s) + tg\big(h(s)\big) \\ &= c^T h(s) + sg\big(h(s)\big) + (t-s)g\big(h(s)\big) \\ &\geq c^T h(t) + sg\big(h(t)\big) + (t-s)g\big(h(s)\big). \end{aligned}$$

Thus $(t-s)g\big(h(t)\big) \geq (t-s)g\big(h(s)\big)$ follows, and proof is completed. □

The last auxiliary tool presents the well-known optimum property of constrained problems with strictly concave objective and linear conditions. For the sake of completeness, we sketch here its proof.

LEMMA 2.4. *Assume that X and Y are Euclidean spaces, $D \subseteq X$ is a nonempty, open, convex set, $A \in \mathcal{L}(X, Y)$, and $b \in Y$. If $f\colon D \to \mathbb{R}$ is a strictly concave function, and the concave program*

$$\begin{aligned} f(x) &\longmapsto \max \\ Ax &= b \end{aligned}$$

has a feasible local maximum, then the local maximum is a global one, and the maximizer is unique.

PROOF. Let $x \in D$ be a feasible local maximizer. Assume to the contrary that the statement is false. Then there exists a feasible solution $y \in D$ such that $f(y) \geq f(x)$. Since x is a local maximizer, it has an open neighborhood $U \subseteq D$, such that $f(x) \geq f(z)$ holds for all $z \in U$ feasible solutions. For $\lambda \in \mathbb{R}$ consider the element $z = \lambda x + (1-\lambda)y$. Then

$$Az = A\big(\lambda x + (1-\lambda)y\big) = \lambda Ax + (1-\lambda)Ay = \lambda b + (1-\lambda)b = b.$$

Thus z is feasible. On the other hand, $z \in U$ with suitable chosen $\lambda \in]0,1[$. The strict concavity and the indirect assumption imply

$$\begin{aligned} f(x) \geq f(z) &= f\big(\lambda x + (1-\lambda)y\big) \\ &> \lambda f(x) + (1-\lambda)f(y) \\ &\geq \lambda f(x) + (1-\lambda)f(x) = f(x), \end{aligned}$$

resulting in the desired contradiction. Hence x is a global feasible maximizer. Uniqueness follows in the same way, using the calculations above and the equality case of the indirect assumption. □

2.2. The main results

The main results are presented in two theorems. The first one gives a sufficient condition for a concave program to have optimal feasible solution. The limit properties involved guarantee that the "large" values of the objective are allocated in a compact subset of the domain. The recession cone plays a key role in the proof.

THEOREM 2.5. *Assume that X and Y are Euclidean spaces, $D \subseteq X$ is a nonempty, open, convex set, $A \in \mathcal{L}(X,Y)$, and $b \in Y$. If a concave function $f\colon D \to \mathbb{R}$ satisfies the limit conditions*

$$\lim_{x \to x_0} f(x) = -\infty \qquad \lim_{\alpha \to +\infty} f(x + \alpha r) = -\infty \tag{2.3}$$

for all $x_0 \in \partial D$ and for all $r \in \operatorname{rec}(D)$, and the concave program

$$\begin{aligned} f(x) &\longmapsto \max \\ Ax &= b \end{aligned}$$

has a feasible solution, then it has a feasible optimal solution, as well.

PROOF. Let $c \in \mathbb{R}$ be given, and consider the sublevel set L_c of the objective function f:

$$L_c(f) := \{x \in D \mid f(x) > c\} = f^{-1}\big(] - \infty, c[\big).$$

We claim that $L_c(f) \subseteq X$ is open, convex and bounded for all $c \in \mathbb{R}$.

The level set is a continuous preimage of an open set, thus it is open in the relative topology of D. However, D itself is open in the Euclidean topology, therefore $L_c(f)$ is open in the Euclidean topology of X. If $x, y \in L_c(f)$ and $\lambda \in [0,1]$, then $\lambda x + (1-\lambda)y \in D$, and

$$f\big(\lambda x + (1-\lambda)y\big) \geq \lambda f(x) + (1-\lambda)f(y) \geq \lambda c + (1-\lambda)c = c.$$

Thus $L_c(f)$ is convex. Finally, assume to contrary that $L_c(f)$ is unbounded. Then there exists a recession direction r by Lemma 1.4. Clearly, r is a recession direction for D, as well. Let $x \in L_c(f)$ be arbitrary. By the second limit condition, there exists $\alpha > 0$ such that $f(x + \alpha r) < c$. On the other hand, the property $x + \alpha r \in L_c(f)$ implies $f(x + \alpha r) > c$, which is obviously impossible.

Now we prove that $\overline{L_c(f)} \subseteq D$. To do this, it suffices to verify that $\partial L_c(f) \subseteq D$. Assume that this is not the case and let $x_0 \in \partial L_c(f) \setminus D$. Then $x_0 \in \partial D$ since $\overline{L_c(f)} \subseteq \overline{D}$ holds evidently. Take a sequence (x_k) in $L_c(f)$ such that $x_k \to x_0$. The inequality $f(x_k) > c$ implies $\lim_{k\to\infty} f(x_k) \geq c > -\infty$, which contradicts to the first limit property.

Closure does not effect convexity and boundedness, therefore $\overline{L_c(f)}$ is a convex, compact set by the Heine–Borel Theorem. The family $\{\overline{L_c(f)} \mid c \in \mathbb{R}\}$ covers D, and we have a feasible solution. Thus $\overline{L_{c_0}(f)}$ intersects the feasible set for some $c_0 \in \mathbb{R}$. The intersection, denoted by K, is compact, being the feasible set closed. By the continuity of f, there exists $x^* \in K$ such that

$$f(x^*) = \max_K f.$$

Now take a feasible solution $x \in D$. If $x \in K$, then $f(x^*) \geq f(x)$ by the choice of x^*. If $x \notin K$, then $x \notin L_{c_0}(f)$, and hence $f(x) \leq c_0$. On the other hand, by continuity again, $f(x^*) \geq c_0$. In other words, x^* is a feasible optimal solution of the concave program. ☐

The second main result is an extension of the log-barrier problem. Observe that the one-parameter family of concave problems reduces to a linear program in standard form if we set $t = 0$. The previous results ensure that its central path exists and it is continuous. Moreover, each limit point of its graph represents an optimal solution of the original primal–dual pair. To check optimality, we do not need second order tests. Thus, we may assume that the objective is *continuously* (instead of *twice*) differentiable. This is the reason why Lagrange multipliers and the Complementary Slackness Theorem are sufficient for us. To formulate the statement, we shall need the concept of standard projections.

DEFINITION. *Let X be a Euclidean space and consider its standard base. If $1 \leq k \leq \dim(X)$, then the k*th standard projection *$\pi_k \colon X \to \mathbb{R}$ is the function which assigns the standard coordinate x_k to all $x \in X$.*

THEOREM 2.6. *Let $D \subseteq \mathbb{R}_+^{n+m}$ be a nonempty, open, convex set, further let $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Assume that $g \colon D \to \mathbb{R}$ is a continuously differentiable strictly concave function such that $\partial_k g$ are nonnegative, $\pi_k \partial_k g$ are bounded, and the function*

$$(x, \omega) \longmapsto c^T x + t g(x, \omega)$$

satisfies the limit conditions (2.3) *for all* $t > 0$. *If the original primal–dual pair has a feasible solution in* D, *then the central path of the family of concave programs*

$$\begin{gathered} c^T x + tg(x, \omega) \longmapsto \max \\ Ax + \omega = b \end{gathered}$$

is well-defined and continuous. Moreover, each limit point at $t = 0$ *of the central path's graph is an optimal solution of the attached primal–dual pair.*

PROOF. For a fixed parameter $t > 0$, the concave program above has a unique optimal solution by Lemma 2.4 and Theorem 2.5. Thus the central path of the family is well-defined, indeed. Taking into consideration the special form of the objective function, the limit conditions (2.3) mean that the map

$$F\bigl(t, (x, \omega)\bigr) := c^T x + tg(x, \omega)$$

fulfills the conditions (2.1) and (2.2). Thus, by Lemma 2.2, the central path is continuous. For the second statement, consider the Lagrange-functional

$$L(x, \omega, y) := c^T x + tg(x, \omega) + y^T (b - Ax - \omega).$$

Then, the unique optimal solution $x = x(t)$ and $\omega = \omega(t)$ of the perturbed problem satisfies the Lagrange system

$$\begin{aligned} \frac{\partial L}{\partial x_j} &= c_j + t\frac{\partial g}{\partial x_j} - \sum_{i=1}^{m} y_i a_{ij} &&= t\frac{\partial g}{\partial x_j} - z_j = 0, \qquad && j = 1, \ldots, n \\ \frac{\partial L}{\partial \omega_i} &= &&= t\frac{\partial g}{\partial \omega_i} - y_i = 0, \qquad && i = 1, \ldots, m. \end{aligned}$$

Since $\partial_k g$ are nonnegative, $y = y(t)$ and $z = z(t)$ are nonnegative, as well. Thus, y is a feasible solution of the dual problem, and z is its slack variable. Moreover, the last terms of the Lagrange system imply

$$t x_j \frac{\partial g}{\partial x_j} = x_j z_j \qquad \text{and} \qquad t\omega_i \frac{\partial g}{\partial \omega_i} = \omega_i y_i.$$

Let $(x^*, \omega^*, y^*, z^*)$ be an arbitrary limit point at $t = 0$ of the primal–dual central path's graph. Since it belongs to $\overline{D}$, it is a feasible solution of the original primal–dual pair. For simplicity, assume that $(x, \omega, y, z)(t) \to (x^*, \omega^*, y^*, z^*)$ as $t \to 0$. Passing the limit $t \to 0$ in the equations above, $0 = x_j^* z_j^*$ and $0 = \omega_i^* y_i^*$ follow for all indices i and j by the boundedness of $\pi_k \partial_k g$. Thus, by Theorem 2.1, $(x^*, \omega^*, y^*, z^*)$ is a feasible optimal solution for the original primal–dual problem.

□

Note also that the generator $F\bigl(t, (x, \omega)\bigr) = c^T x + tg(x, \omega)$ fulfills the conditions of Lemma 2.3. Thus the values of the objective increases along the central path as the parameter t approaches to zero.

2.3. Applications

As a direct application of our previously presented main results, we revisit the classical log-barrier problem and handle it as a direct corollary of Theorem 2.6. Consider a linear program in standard form, of which constraints are given in equality form using slack variables. Requiring strict positivity on all the variables, perturb the original objective function for all parameters $t > 0$ in the following way:

$$\begin{aligned} c^T x + t\sum_{j=1}^{n} \log x_j + t\sum_{i=1}^{m} \log \omega_i &\longmapsto \max \\ Ax + \omega &= b \\ x, \omega &> 0. \end{aligned}$$

This one-parameter family of constrained programs is the classical *log-barrier problem.*

COROLLARY 2.7. *If the original primal–dual pair has a positive feasible solution, then the central path of the log-barrier problem exists and it is continuous. Moreover, its each limit point at zero is a feasible optimal solution of the original primal–dual pair.*

PROOF. We show that the conditions of Theorem 2.6 hold. Let $D \subseteq \mathbb{R}_+^{n+m}$ be the positive orthant. Then, D is a nonempty, open and convex set. For all $k = 1, \dots, n+m$ define $\varphi_k \colon D \to \mathbb{R}$ by

$$\varphi_k(x, \omega) := \begin{cases} \log x_j & \text{if } k = j; \\ \log \omega_i & \text{if } k = n+i, \end{cases}$$

where $j = 1, \dots, n$ and $i = 1, \dots, m$; moreover, let $g \colon D \to \mathbb{R}$ be defined by

$$g(x, \omega) := \sum_{k=1}^{n+m} \varphi_k(x, \omega).$$

Then, g is continuously differentiable and strictly concave; its partial derivatives are positive and $\pi_k \partial_k g = 1$. The objective function of the barrier-problem can be written as

$$f(x, \omega) = c^T x + t g(x, \omega).$$

Finally, we prove that f satisfies the limit conditions (2.3). Let (x_*, ω_*) be a positive feasible solution of the primal, and (y_*, z_*) be a positive feasible solution of the dual. Using the definition of slack variables,

$$z_*^T x + y_*^T \omega = (A^T y_* - c)^T x + y_*^T (b - Ax) = y_*^T b - c^T x$$

holds for each primal feasible solution (x, ω). Thus,

$$\begin{aligned} f(x,\omega) &= t \sum_{k=1}^{n+m} \varphi_k(x,\omega) + c^T x \\ &= t \sum_{k=1}^{n+m} \varphi_k(x,\omega) + y_*^T b - z_*^T x - y_*^T \omega. \end{aligned}$$

If $(x_0, \omega_0) \in \partial D$ and $(x, \omega) \to (x_0, \omega_0)$, then there exists an index j such that $x_j \to 0$, or there exists an index i such that $\omega_i \to 0$. Then $\varphi_k(x, \omega) \to -\infty$ where k corresponds to j or i, while $c^T x$ has a finite limit. Thus, in this case, $f(x, \omega) \to -\infty$.

If $(x, \omega) \in \operatorname{rec}(D)$, then $x, \omega > 0$. Using the convention $x_{n+i} = \omega_i$ again, the objective function can be represented as the sum of terms

$$t \log(\alpha x_k) - \alpha x_k + \beta.$$

Here x_k are positive therefore $f(\alpha x, \alpha\omega) \to -\infty$ as $\alpha \to +\infty$. Thus the statement follows directly from Theorem 2.6. □

The boundedness of the functions $\pi_k \partial_k g$ implies that the choice of the logarithm function is optimal among a certain class of barriers in the following sense. For all $i \in \{1, \ldots, m\}$ denote the components of the slack variable ω by $x_{n+i} := \omega_i$ and assume that $g\colon \mathbb{R}_+^{n+m} \to \mathbb{R}$ has the sum form

$$g(x) := \sum_{k=1}^{n+m} g_k(x_k).$$

Here the functions $g_k\colon]0, +\infty[\to \mathbb{R}$ are continuously differentiable, strictly concave functions such that $\partial_k g_k$ are nonnegative according to the assumptions of Theorem 2.6.

For a fixed $k \in \{1, \ldots, n+m\}$ assume that $C_1 \leq \pi_k \partial_k g \leq C_2$. Since $\partial_k g = g_k'$ and $x_k > 0$ we can rewrite these inequalities in the following form:

$$\frac{C_1}{x_k} \leq g_k'(x_k) \leq \frac{C_2}{x_k}.$$

Thus $C_1 \log(u) \leq g_k(u) \leq C_2 \log(u)$. This phenomena explains the distinguished role of the perturbing functions in the classical barrier problem.

CHAPTER 3

An optimality condition for linear programs

In this chapter we are going to focus on the geometrical properties of linear programs and give a geometrical optimality condition as a main result. To prove it, we use the standard tools of Convex Geometry and Convex Analysis combined with induction. As applications, we present a geometric proof of the strong duality theorem and revisit Farkas' lemma.

Let us emphasize that many auxiliary tools are well-known or direct consequences of classical and highly nontrivial results. However, we present their proof for two reasons: In order to make the exposition self-contained and to use exactly just the needed background. We hope that the present chapter in this way may have didactic impacts during lectures on the topic.

3.1. Auxiliary tools

Firstly, let us introduce the forms of linear programs which we are going to work with throughout this chapter. The basic one is

$$\begin{aligned} \langle c, x\rangle &\longmapsto \max \\ x &\in P \end{aligned} \tag{3.1}$$

where $c \in \mathbb{R}^n$ is a given vector and $P \subseteq \mathbb{R}^n$ is a polyhedron. As usual, we can describe the polyhedron P as a mixed system of equalities and inequalities. Moreover, to each problem we can attach an other one in a well-motivated way and then we can speak about the *primal–dual pair* of linear programs. Throughout this chapter we shall use the primal in *canonical form* while the dual in *standard form* in the following way:

$$\text{Primal:}\quad \begin{aligned} \langle c, x\rangle &\longmapsto \max \\ Ax &= b \\ x &\geq 0; \end{aligned} \qquad\qquad \text{Dual:}\quad \begin{aligned} \langle y, b\rangle &\longmapsto \min \\ A^T y &\geq c \end{aligned} \tag{3.2}$$

where $A \in \mathbb{R}^{m\times n}$ is a matrix with m rows and n columns, further $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are given vectors.

The feasible sets of linear programs in both canonical and standard forms are convex. Their recession cones can be directly described as the next result shows.

LEMMA 3.1. *If* $A \in \mathbb{R}^{m\times n}$ *and* $b \in \mathbb{R}^m$ *then*

$$\operatorname{rec}\{x \in \mathbb{R}^n \mid Ax \le b\} = \{x \in \mathbb{R}^n \mid Ax \le 0\}.$$

In particular, the recession cone of the set $\{x \in \mathbb{R}^n \mid Ax = b\}$ *coincides to the kernel of A.*

PROOF. If $x + \alpha r$ belongs to $\{x \in \mathbb{R}^n \mid Ax \le b\}$ for all $\alpha \ge 0$, then $Ax + \alpha Ar = A(x + \alpha r) \le b$. Thus Ar cannot have any positive component. Conversely, if Ar has only nonpositive components and $\alpha \ge 0$ then

$$A(x + \alpha r) = Ax + \alpha Ar \le Ax \le b$$

whenever $Ax \le b$. Therefore r is a recessional direction. The second statement is a direct consequence of the first one. □

Applying the basic separation property of Lemma 1.1 and Lemma 1.2 we can prove an important connection between the primal and the dual problem. The key ideas can be adopted directly from the book of Barvinok.

LEMMA 3.2. *If a primal problem has an optimal solution, then the dual is feasible.*

PROOF. For simplicity we focus only on primal–dual pairs given in the form (3.2). Consider the set

$$K := \{(Ax, \langle c, x\rangle) \in \mathbb{R}^{m+1} \mid x \in \mathbb{R}^n_+\}.$$

Then K is a cone evidently; moreover, it is finitely generated and hence it is closed by Lemma 1.2. Assume that x^* is an optimal primal solution and let $\gamma := \langle c, x^*\rangle$. If $\delta > \gamma$ is a fixed real number, then $(b, \delta) \notin K$. Thus, by Lemma 1.1, there exist a pair $(y, \mu) \in \mathbb{R}^{m+1}$ and $\lambda \in \mathbb{R}$ such that

$$\langle b, y\rangle + \mu\delta < \lambda \qquad \text{and} \qquad \langle Ax, y\rangle + \mu\langle c, x\rangle > \lambda.$$

Substituting $x = 0$ we get $\lambda < 0$. On the other hand, K is a cone thus the latter inequality yields

$$\langle Ax, y\rangle + \mu\langle c, x\rangle \ge 0$$

whenever $x \ge 0$. In particular, the choice $x = x^*$ and the facts above result in

$$\langle b, y\rangle + \mu\delta < \lambda \le 0 \le \langle Ax^*, y\rangle + \mu\langle c, x^*\rangle = \langle b, y\rangle + \mu\gamma.$$

Therefore $0 < \mu(\gamma - \delta)$ and hence $\mu < 0$. Without the loss of generality we may assume that $\mu = -1$. Then

$$0 \le \langle Ax, y\rangle - \langle c, x\rangle = \langle x, A^T y\rangle - \langle c, x\rangle = \langle x, A^T y - c\rangle$$

for all $x \ge 0$. Thus $A^T y \ge c$ showing that y is a dual feasible solution. □

Finally, we shall need an other special type of cones, namely the normal cone of a convex set. In the geometric point of view, the normal cone consists of those vectors which make an obtuse angle with vectors initiating from p and terminating in the points of the set.

DEFINITION. *If D is a convex subset in a vector space X and $p \in D$, then the* normal cone *of D at the point p is the set*

$$\mathcal{N}_D(p) := \{y \in X \mid \langle y, x - p\rangle \leq 0,\, x \in D\}.$$

3.2. The main result

Now we are in the position to formulate and prove our main result of this chapter.

THEOREM 3.3. *The linear program* (3.1) *has an optimal solution if and only if it is feasible and*

$$c \in \mathcal{N}_{\mathrm{rec}(P)}(0).$$

PROOF. Consider the linear program (3.1), where $P \subseteq \mathbb{R}^n$ is a polyhedron and $c \in \mathbb{R}^n$. From now on, we may assume that $c \neq 0$ (otherwise the statement is obvious).

If (3.1) has an optimal solution x^*, then the program is clearly feasible. If $r \in \mathrm{rec}(P)$, then the element $x^* + \alpha r$ is feasible for all $\alpha \geq 0$. By optimality,

$$\langle c, x^*\rangle \geq \langle c, x^* + \alpha r\rangle = \langle c, x^*\rangle + \alpha\langle c, r\rangle$$

yielding the desired condition $\langle c, r\rangle \leq 0$.

We prove the converse statement by induction on the dimension n. For $n = 1$, the feasible set P is a (not necessarily bounded) closed interval, which is nonempty by assumption. Moreover, the obtuse angle property and the technical assumption $c \neq 0$ exclude the case $P = \mathbb{R}$. The remaining possibilities can easily be checked; we omit the details.

Assume that the statement is true in $\mathbb{R}^n$ for *all* programs. Consider a feasible linear program fulfilling the obtuse angle condition in $\mathbb{R}^{n+1}$. Then necessarily $c \notin \mathrm{rec}(P)$. Thus if $x_0 \in P$, then there exists $\alpha \geq 0$ such that $x_0 + \alpha c \in \partial P$. Since P is the intersection of half-spaces in $\mathbb{R}^{n+1}$, there exists a facet $\mathcal{F} \subseteq P$ such that $x_0 + \alpha c \in \mathcal{F}$. Denote the orthogonal projection of c on one of the hyperplane that induces the facet $\mathcal{F}$ by v and let $w := c - v$. Then w is a normal vector of $\mathcal{F}$. Now define an auxiliary program by

$$\langle v, x\rangle \longmapsto \max \qquad \text{subject to} \qquad x \in \mathcal{F}.$$

We may consider this program as an n-dimensional one which is clearly feasible. On the other hand, $\mathrm{rec}(\mathcal{F}) \subseteq \mathrm{rec}(P)$ by Lemma 3.1. Thus, for all $r \in \mathrm{rec}(\mathcal{F})$ we arrive at

$$0 \geq \langle c, r\rangle = \langle v + w, r\rangle = \langle v, r\rangle.$$

Therefore, the auxiliary program fulfills the obtuse angle condition and hence it has an optimal solution x^* by assumption. We claim that x^* is optimal on $\mathcal{F}$ even for the original objective. Indeed, for all $x \in \mathcal{F}$ we have

$$\begin{aligned}\langle c, x\rangle &= \langle v + w, x\rangle = \langle v, x\rangle + \langle w, x\rangle \leq \langle v, x^*\rangle + \langle w, x\rangle \\ &= \langle c - w, x^*\rangle + \langle w, x\rangle = \langle c, x^*\rangle + \langle w, x - x^*\rangle = \langle c, x^*\rangle,\end{aligned}$$

since $x - x^*$ belongs to the orthogonal complement of w. Now consider the modified version of the original program given by

$$\langle c, x\rangle \longmapsto \max \quad \text{subject to} \quad x \in P_1 := P \cap \{x \in \mathbb{R}^{n+1} \mid \langle c, x\rangle \geq \langle c, x^*\rangle\}.$$

Clearly, this is feasible and the original obtuse angle condition remains true. Moreover, the original problem has an optimal solution if and only if this modified problem has an optimal solution. If

$$P \cap \{x \in \mathbb{R}^{n+1} \mid \langle c, x\rangle > \langle c, x^*\rangle\} = \emptyset,$$

then $x_1^* := x^*$ is an optimal solution to the modified (equivalently: to the original) program. If this is not the case, then repeat the previous process with the modified program. This way we can recursively define a sequence of linear programs

$$\langle c, x\rangle \longmapsto \max \quad \text{subject to} \quad x \in P_{k+1} := P_k \cap \{x \in \mathbb{R}^{n+1} \mid \langle c, x\rangle \geq \langle c, x_k^*\rangle\}.$$

To complete the proof we show that (P_k) terminates in finite steps. Observe that the number of facets in P_1 does not exceed the number of facets in P. Indeed, the half-space in the intersection may create a new facet but the essential part of $\mathcal{F}$, that is, the set

$$\mathcal{F} \cap \{x \in \mathbb{R}^{n+1} \mid \langle c, x\rangle < \langle c, x^*\rangle\}$$

completely disappears. Moreover, from the second recursion step the number of facets strictly decreases since the boundary of the half-spaces in the intersections are parallel. On the other hand, a polyhedron has only finite number of facets and thus the process terminates in finite steps. □

3.3. Applications

Using the main result above, we can give the optimality condition for problems in canonical or standard form immediately.

COROLLARY 3.4. *A maximum problem in canonical form has an optimal solution if and only if it is feasible and* $\langle c, r\rangle \leq 0$ *for all* $r \in \mathrm{Ker}(A) \cap \mathbb{R}^n_+$.

PROOF. Let us consider the maximum problem in canonical form from (3.2). Firstly, assume that it has an optimal solution. Then, it is clearly feasible, that is, the feasible set

$$P := \{x \in R^n \mid Ax = b, x \geq 0\}$$

is nonempty. Since both sets $\{x \in \mathbb{R}^n \mid Ax = b\}$ and $\{x \in \mathbb{R}^n \mid x \geq 0\}$ are convex and closed, then P is a convex, closed subset of both of them. Thus the recession cone of P can be calculated as

$$\begin{aligned}\mathrm{rec}(P) &= \mathrm{rec}\big(\{x \in \mathbb{R}^n \mid Ax = b\} \cap \{x \in \mathbb{R}^n \mid x \geq 0\}\big)\\ &= \mathrm{rec}\big(\{x \in \mathbb{R}^n \mid Ax = b\}\big) \cap \mathrm{rec}\big(\{x \in \mathbb{R}^n \mid x \geq 0\}\big) = \mathrm{Ker}(A) \cap \mathbb{R}^n_+\end{aligned}$$

by Lemma 1.6 and Lemma 3.1, and then the second part of the statement follows from Theorem 3.3.

The converse statement can be proved using Lemma 1.6, Lemma 3.1 and Theorem 3.3 along with the previous calculations but in the opposite direction. □

COROLLARY 3.5. *A minimum problem in standard form has an optimal solution if and only if it is feasible and* $\langle b, q\rangle \geq 0$ *whenever* $A^T q \geq 0$.

PROOF. Consider the minimum problem in standard form from (3.2) and rewrite the problem as a maximum one:

$$-\langle y, b\rangle \longmapsto -\max$$
$$-A^T y \leq -c.$$

Let us introduce the notations $M := -A^T$ and $v := -c$. Assume that it has an optimal solution. Thus it is feasible as well, that is, the feasible set

$$P := \{y \in R^m \mid My \leq v\}$$

is nonempty. Then its recession cone can be calculated as

$$\begin{aligned}\mathrm{rec}(P) &= \mathrm{rec}\big(\{y \in \mathbb{R}^m \mid My \leq v\}\big)\\ &= \{y \in \mathbb{R}^m \mid My \leq 0\} = \{y \in \mathbb{R}^m \mid -A^T y \leq 0\}\end{aligned}$$

by Lemma 3.1, and then the second part of the statement follows from Theorem 3.3.

The converse statement can be proved using Lemma 3.1 and Theorem 3.3 along with the previous calculations but in the opposite direction. □

The strong duality theorem is one of the most important cornerstone of linear programming. Vanderbei proves it with the help of the simplex algorithm. We present here an independent approach based on the basic separation theorem and our main result.

COROLLARY 3.6. *If a primal problem has an optimal solution, then so does its dual.*

PROOF. We may assume that the primal is a maximum problem in canonical form and the dual is a minimum problem in standard form. If the primal has an optimal solution, then the dual is feasible by Lemma 3.2. If $x^* \geq 0$ is an optimal primal solution and q satisfies $A^T q \geq 0$, then

$$\langle b, q\rangle = \langle Ax^*, q\rangle = \langle x^*, A^T q\rangle \geq 0.$$

This property completes the proof by Corollary 3.5. □

In fact, the strong duality theorem is equivalent to Farkas' lemma. This fact allows a geometric approach to duality theory in linear programming besides the algorithmic ones. Similarly to Vanderbei, we deduce Farkas' lemma from a suitable primal–dual pair. However, we completely avoid the simplex algorithm.

COROLLARY 3.7. *If* $A \in \mathbb{R}^{m\times n}$ *and* $b \in \mathbb{R}^m$ *then*

(1) either the system $Ax = b$ *and* $x \geq 0$ *has a solution;*
(2) or the system $A^T y \geq 0$ *and* $b^T y < 0$ *has a solution.*

PROOF. Consider the primal–dual pair (3.2) with $c = 0$. The solvability of (1) means that the primal is feasible. Now this fact is equivalent to the optimality of the primal since $c = 0$. Corollary 3.6 ensures that the dual is optimal, as well. However, the dual is evidently feasible; thus it is optimal if and only if $b^T y \geq 0$ whenever $A^T y \geq 0$ by Corollary 3.5. In other words, system (2) has no solution. □

We mention here that the proof of Theorem 3.3 suggests an algorithm. Our process moves on facets (and not between vertices) and hence it may differ from the usual simplex method. The exact connection of these methods however, is still an open problem. Furthermore, Corollary 3.4 and Corollary 3.5 may be applied to quickly check whether a linear program has an optimal solution or not.

Normal cones play a central role in convex programming. Let us quote here the well-known first order optimality condition (for details, see Borwein and Lewis). Suppose that $D \subseteq \mathbb{R}^n$ is convex and $f\colon D \to \mathbb{R}$ is convex and differentiable. Then a point x^* is a global maximizer if and only if $f'(x^*) \in \mathcal{N}_D(x^*)$. In case of linear programs, $c = f'(x^*)$ is independent on x^* while $\mathcal{N}_D(x^*)$ is not. That is, to check optimality we have to find first a candidate for the stationary point. However, Theorem 3.3 does not require such *á priori* information thus it seems to be more suitable for linear programs than the mentioned one.

Finally, we mention that the main result may follow from the decomposition theorem by Motzkin, stating that a polyhedron in a Euclidean space can be represented as the Minkowski sum of a polytope and a polyhedral cone. However, this result is highly nontrivial: As a consequence of the fact that the concepts of "polyhedral" and "finitely generated" are equivalent for cones, it relies on the fundamental theorem of linear inequalities obtained by Farkas [**11**], Minkowski [**22**] and Weyl [**31**]. Moreover, some proofs of the Farkas–Minkowski–Weyl Theorem use the simplex method; see for example the book of Schrijver which also serves as an excellent reference of the topic. About our opinion, these facts justify our independent and self-contained approach to Theorem 3.3.

CHAPTER 4

Computer implementation for testing solvability of linear programs

Considering everyday life problems it may occur that only the solvability issue of a linear program counts in a practical application and the solution itself (if it exists) does not. Motivated by this phenomena and the main result in CHAPTER 3, in the following part we suggest an implementable method for testing the solvability of linear programs (different from the one that was suggested by the proof of the main result in the previous chapter). The main aim of this chapter is to present an algorithm for the planar case, validate it and then discuss its "imperfections" and the possibilities for the higher-dimensional versions.

4.1. Auxiliary tools

To be able to create our implementable algorithm, the polyhedron P in the constraints should be given by linear inequalities. Therefore, let our 2-dimensional linear programming problem be given in the form

$$\begin{aligned} c_1x_1 + c_2x_2 &\longmapsto \max \\ a_{i1}x_1 + a_{i2}x_2 &\leq b_i, \end{aligned} \tag{4.1}$$

where $c_j, a_{ij}, b_i \in \mathbb{R}$ are given for all $j \in \{1, 2\}$ and $i \in \{1, \dots, m\}$ indices.

Let us mention that in this chapter we will use the representation of vectors where the base point is the origin, so it is enough to give the coordinates of their endpoints. This allows us to see vectors as points of the plane, therefore, we are going to use these concepts interchangeably since it will not cause any misunderstandings.

We say that if $p \in \mathbb{R}^2$, then p and $-p$ are *opposing pair of points*. Furthermore, we call the vector v of a convex cone an *extremal* vector if the cone remains convex after excluding v from it.

Finally, let us also introduce some further notations we are going to use in the remaining part of this chapter:

- $[p_1, \dots, p_k]$: ordered list of the points $p_1, \dots, p_k$;
- $[\,]$: empty list;
- $|L|$: number of elements of the list L;
- $\mathcal{N}$: normal cone of the feasible set.

Now we are in the position to formulate our method.

4.2. The method

Algorithm 4.1 makes an equivalence classification on the normal vectors (a_{i1}, a_{i2}) which determine the feasible set such that it norms these vectors to the unit circle of $\mathbb{R}^2$. Then, it collects one representative of each class into the reduced point list R.

Algorithm 4.1 *Point list reduction*

```
Input: Normal vectors $(a_{i1}, a_{i2})$ for all $i \in \{1, \dots, m\}$
Output: Reduced point list $R$
    (1) Collect the normal vectors $(a_{i1}, a_{i2})$ into the list $L$ such that the points
        are normalized to the unit circle of $\mathbb{R}^2$.
    (2) If $|L| = 1$ then $R := L$.
    (3) Else collect one representative of each equivalence class into $R$.
```

From the reduced point list R that Algorithm 4.1 provided we need to collect all opposing pair of points because they can either induce special types of cones, or make the decision on the separation of points easier.

Algorithm 4.2 *Collecting opposing pair of points*

```
Input: Reduced point list $R$
Output: Point-pair list $N$ containing opposing pair of points from $R$
    (1) If $|R| = 1$ then $N := [\,]$.
    (2) If $|R| > 1$ then check $R$:
        (a) If there are opposing pair of points in $R$ then collect them pairwise
            in $N$.
        (b) Else $N := [\,]$.
```

Before discussing the final part, we need to define the separation property of a set of planar points. We say that the line ℓ *separates* the points of a planar set R if both open half-planes (determined by ℓ) intersect R. Let us introduce the logical function $\text{Sep}\colon R \to \{\texttt{TRUE}, \texttt{FALSE}\}$ to be able to use this property in a more convenient way later. Thus let

$$\text{Sep}(p) := \begin{cases} \texttt{TRUE} & \text{if } \ell \text{ separates the points in } R \\ \texttt{FALSE} & \text{otherwise,} \end{cases}$$

where ℓ is the attached line to the point $p \in R$.

Now we are in the position to formulate our main algorithm to decide on the solvability of (4.1). During the steps, we are going to indicate with a `STOP` sign when Algorithm 4.3 has its output value.

Algorithm 4.3 *Calculating the normal cone*

```
Input: Reduced point list $R$ and list of opposing pair of points $N$
Output: List of extremal vectors $E$ that generate the normal cone $\mathcal{N}$ of the feasible set
  (1) If $|N| \geq 2$ then $E := [\,]$ and STOP.
  (2) If $|N| = 1$ then $N = \big[[p, -p]\big]$ and $\ell$ is the attached line to $p$ and $-p$.
      Check Sep$(p)$:
      (a) If Sep$(p) =$ FALSE and
          (i) $|R| = 2$ then $E := R$ and STOP.
          (ii) $|R| > 2$ then $E := [p, -p, q]$ where $q \in R\backslash\{p, -p\}$ and STOP.
      (b) Else $E := [\,]$ and STOP.
  (3) If $|N| = 0$ and
      (a) $|R| = 1$ then $E := R$ and STOP.
      (b) $|R| > 1$ then search for extremals:
          For $i = 1, \ldots, |R|$:
          Pick $p_i \in R$ and the attached line $\ell(p_i)$.
          Check Sep$(p_i)$:
            (i) If Sep$(p_i) =$ FALSE then add $p_i$ to $E$ and continue the cycle.
            (ii) Else continue the cycle.
          Return $E$ and STOP.
```

THEOREM 4.1. *The previously presented method is mathematically correct and it stops in finitely many steps. Moreover, if the problem is feasible and at the end of Algorithm 4.3*

(1) E is empty, then there exists an optimal solution;
(2) E is nonempty and $c \in \operatorname{cone}(E)$, then there exists an optimal solution, otherwise not.

In particular, if $c \notin \operatorname{cone}(E)$, then there is no optimal solution.

PROOF. Without the loss of generality we may assume that the problem is feasible and we have the lists R, N and E.

If $|N| \geq 2$, then we can decompose the plane to 4 conics such that their vertices are in the origin, their angle are convex, and their generators are determined by 2 pairs of N. Thus any point of the plane belongs to one of the conics, yielding that the normal cone is the plane. Therefore, the feasible set is compact and hence the problem is solvable.

If $|N| = 1$, then consider the line ℓ determined by N. If it separates R then the normal cone is the plane, and hence the problem is solvable. If not, then at most one of the open half-planes (determined by ℓ) may intersect R. If this occurs, then the closure of the intersecting half-plane is the normal cone, otherwise the line ℓ. In both cases, Theorem 3.3 decides solvability.

If $|N| = 0$, then any line passing trough the origin contains at most 1 point of R. If each of these lines separates R, then the normal cone is the plane, and hence the problem is solvable. If this is not the case, consider a line $\ell(p)$ determined by p that does not separate R. Then at most one of the obtained open half-planes may intersect R. If this does not occur, then $E = [p]$. If it does, then there exists a point $q \in R$ that makes a maximal angle with p. (Note that R is finite.) The points p and q are on the same open half-plane, hence the line $\ell(q)$ determined by q cannot separate R. (Otherwise the angle would not be maximal). In this case, $E = [p, q]$, and in each case, Theorem 3.3 decides solvability.

Finally, the algorithm stops in finite steps. Indeed, if $|N| \geq 1$, then the finiteness of R itself guarantees this claim; if $|N| = 0$, then the "`FOR`" cycle of the algorithm detects p and q avoiding infinite cycling. $\square$

One of the disadvantages of this algorithm is that it cannot handle the feasibility issue of a given linear program. This is due to the fact that the feasibility highly depends on the constants b_i in the constraint system, but the algorithm does not. In the geometrical sense, changing the values of b_i means that the borderlines of each attached half-planes are translated. Thus if we choose these constants appropriately, then we can make any feasible set nonempty. However, this freedom is usually not granted in practice.

As for the higher-dimensional version, considering this method and the previous proof, some problems might occur. One problematic point is checking the separation property. If the dimension of the problem is n, then we need to use $(n - 1)$-dimensional subspaces for the separation. However, even in the $n = 3$ case the "maximum angle" of two points cannot be applied in such way as we did it in the 2-dimensional case. Thus the final part of our proof needs to be treated in other ways.

Finally, let us mention that the number of computational steps can be reduced when implementing the 2-dimensional case. If the "`FOR`" cycle finds one extremal, then it only needs to run until it finds the other one since in $\mathbb{R}^2$ a convex cone (if it is not a half-plane) can be uniquely generated by at most 2 extremal vectors. However, in higher-dimensions all given points might be extremals for the normal cone, thus the "`FOR`" cycle cannot have such stopping condition. Therefore, considering the amount of the required calculations, this method not necessarily has advantage in practice over other type of algorithms apart from the 2-dimensional version.

As a last part of this section we are going to show how the previous method works via three examples. Since our method requires the constraint system to be in the form of $a_{i1}x_1 + a_{i2}x_2 \leq b_i$, for simplicity, the examples will be already given like that. Furthermore, we are going to attach some figures for each problem which were generated by the computer implementation in Maple based on our method (see the Maple codes in the Appendix).

EXAMPLE 4.2. *Decide on the solvability of the following linear programming problem:*

$$
\begin{aligned}
3x_1 + 2x_2 &\longmapsto \max \\
2x_1 + x_2 &\leq 10 \\
x_1 + x_2 &\leq 8 \\
x_1 &\leq 4 \\
-x_1 &\leq 0 \\
-x_2 &\leq 0.
\end{aligned}
$$

SOLUTION: The initial data for Algorithm 4.3:

$$
\begin{aligned}
c &= (3, 2) \\
L &= \big[(2,1), (1,1), (1,0), (-1,0), (0,-1)\big] \\
R &= L \\
N &= \big[[(1,0), (-1,0)]\big] \\
E &= [\,]
\end{aligned}
$$

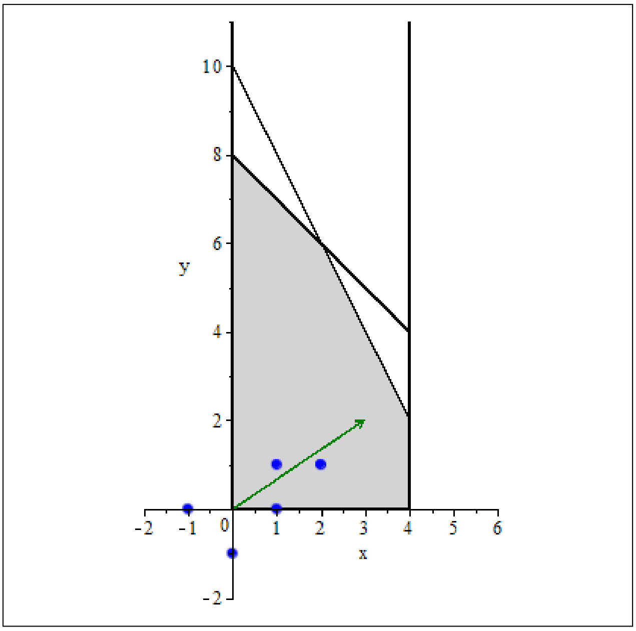


FIGURE 4.1. The gray area represents the feasible set using the original constraint system; the blue dots represent the original normal vectors (a_{i1}, a_{i2}); the green vector is the coefficient vector c of the objective function.

$|N| = 1$ means that Algorithm 4.3 starts with **Step (2)**. The attached line to the opposing pair of points $[(1,0),(-1,0)]$ is the x-axis. Since there are points on both side of the line Algorithm 4.3 stops at **Step (2/b)**. Thus, E being empty results that the normal cone $\mathcal{N}$ is the whole plane, so the feasible set is compact. Observe on Figure 4.1 that the feasible set is nonempty, thus the problem has an optimal solution by Theorem 4.1. $\square$

EXAMPLE 4.3. *Decide on the solvability of the following linear programming problem:*

$$\begin{aligned} -2x_1 + x_2 &\longmapsto \max \\ -3x_1 - 3x_2 &\leq 6 \\ -3x_1 &\leq 0 \\ -x_1 - x_2 &\leq 2 \\ 2x_1 + 2x_2 &\leq 4. \end{aligned}$$

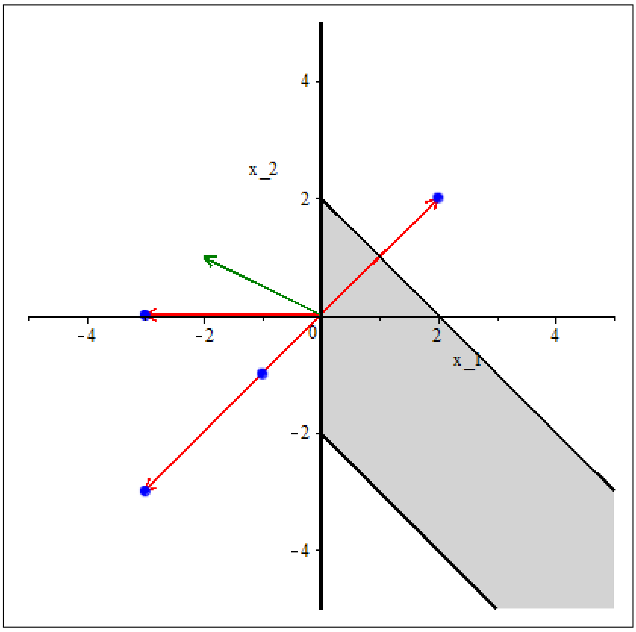


FIGURE 4.2. The gray area represents the feasible set using the original constraint system; the blue dots represent the original normal vectors (a_{i1}, a_{i2}); the red vectors are the elements of E which generate the normal cone $\mathcal{N}$; the green vector is the coefficient vector c of the objective function.

SOLUTION: The initial data for Algorithm 4.3 is

$$\begin{aligned} c &= (-2,1) \\ L &= \big[(-3,-3),(-3,0),(-1,-1),(2,2)\big] \\ R &= \big[(-3,-3),(-3,0),(2,2)\big] \\ N &= \big[[(-3,-3),(2,2)]\big] \\ E &= R \end{aligned}$$

Since $|N| = 1$ and $|R| = 3$ Algorithm 4.3 starts at **Step (2)** again but this time it stops at **Step (2/a/ii)**. Thus the normal cone $\mathcal{N}$ is a half-plane. On Figure 4.2 the green vector clearly lies in the half-plane determined by the red vectors which means that $c \in \mathcal{N}$. The feasible set is also nonempty, thus the problem has an optimal solution by Theorem 4.1. ☐

EXAMPLE 4.4. *Decide on the solvability of the following linear programming problem:*

$$\begin{aligned} -4x_1 - x_2 &\longmapsto \max \\ 2x_1 + 8x_2 &\le 7 \\ 2x_1 + 2x_2 &\le -6 \\ 3x_1 &\le 4 \\ 4x_1 - 2x_2 &\le 8 \\ 4x_1 + 4x_2 &\le -4 \\ 6x_1 &\le 16 \\ 8x_1 - 4x_2 &\le 8 \\ x_1 + 2x_2 &\le 2 \\ 6x_1 + 6x_2 &\le 12. \end{aligned}$$

SOLUTION: The initial data for Algorithm 4.3 is

$$\begin{aligned} c &= (-4,-1) \\ L &= \big[(2,8),(2,2),(3,0),(4,-2),(4,4),(6,0),(8,-4),(1,2),(6,6)\big] \\ R &= \big[(2,8),(2,2),(3,0),(4,-2),(1,2)\big] \\ N &= [\,] \\ E &= \big[(2,8),(4,-2)\big] \end{aligned}$$

Since $|N| = 0$ and $|R| = 5$ Algorithm 4.3 runs the `FOR` cycle in **Step (3/b)** until it finds 2 extremals. Then, observe on Figure 4.3 that the green vector does not lie in the convex cone determined by the red extremal vectors of the normal cone $\mathcal{N}$. Thus the objective function is unbounded which means that the problem has no optimal solution by Theorem 4.1. ☐

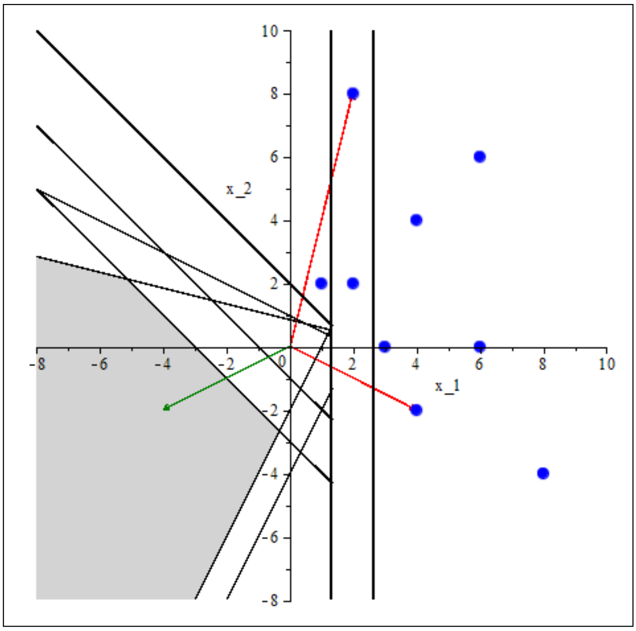


FIGURE 4.3. The gray area represents the feasible set using the original constraint system; the blue dots represent the original normal vectors (a_{i1}, a_{i2}); the red vectors are the elements of E which generate the normal cone $\mathcal{N}$; the green vector is the coefficient vector c of the objective function.

Summary

The motivation of the present dissertation is to revisit one of the interior point methods, namely the log-barrier problem, which has been proved to have polynomial complexity. Moreover, the geometrical properties of linear programs allow us to develop geometrical conditions and new type of methods for these problems, as well. Our aim is to approach linear programs via the tools of Convex Analysis and Convex Geometry while we avoid the usage of the simplex method. The chapters, in turn, are based on the materials of the papers [**5**], [**6**].

In CHAPTER 1 we collect the basic tools of Convex Analysis and Convex Geometry that we use generally or in most of the other chapters such as the basic definition of convex and concave functions; the Bernstein–Doetch Theorem; the definition of convex and conic hulls of sets. Further, we derive linear programs as a special case of general constrained optimization problems.

The basic separation theorem (see Lemma 1.1) plays a crucial role in optimization and several proofs are known for its Euclidean version. In this chapter we prove this statement following the alternative approach suggested by Barvinok.

Let $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ stands for the vector space of all linear maps from $\mathbb{R}^n$ to $\mathbb{R}^m$. Linear transformations $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ may destroy the closedness of (convex) sets even in very simple cases. However, if the set is a polyhedron, then its linear transform still remains a polyhedron. The proof of this fact is surprisingly complicated, but fortunately, we can avoid inconvenience with the next observation (Lemma 1.2).

LEMMA. *Any finitely generated cone of a vector space is the union of the independently generated subcones. In particular, any finitely generated cone of a finite dimensional space is closed.*

The proof of this statement relies on the conical Carathéodory theorem.

The recessional directions of a convex subset D of a vector space play crucial roles in the success of developing our goals. Rockafellar investigated the case when D is a *closed*, convex set, and it turns out that the recessional directions of D have many important properties. They are independent on the choice of $x \in D$ and the existence of recessional directions characterize the unboundedness of D. Furthermore, the recessional directions form a cone: they are closed on addition and multiplication with nonnegative real numbers. Thus, the set of all recessional directions denoted by $\operatorname{rec}(D)$ is called the *recession cone* of D. However, the objective's domain of the log-barrier problem requires us to use *open*, convex sets. But it turns out that the same statements remain true in this case too (Lemma 1.4).

LEMMA. *If D is a nonempty, open, convex subset of a Euclidean space X, and $r \in X$, then the next statements are equivalent:*

(1) There exists $x \in D$, such that $x + \alpha r \in D$ for all $\alpha \geq 0$;
(2) For all $x \in D$ and for all $\alpha \geq 0$, we have that $x + \alpha r \in D$.

Moreover, D is bounded if and only if $\operatorname{rec}(D)$ *is trivial.*

Despite the similar properties of the two cases, we need to follow an alternative approach in the proof.

Finally, we mention a calculus property of recession cones which holds in both the open and closed cases (Lemma 1.5 and Lemma 1.6). In fact, the reasoning in their proof is also very similar.

LEMMA. *If A and B are intersecting open/closed, convex subsets of a Euclidean space X, then*

$$\operatorname{rec}(A \cap B) = \operatorname{rec}(A) \cap \operatorname{rec}(B).$$

The aim of CHAPTER 2 is to achieve an extension of the classical log-barrier problem while avoiding the usage of the simplex method and also correcting Vanderbei's mistake. For this, we work with the particular form of linear programs

$$\text{Primal:} \quad \begin{aligned} c^T x &\longmapsto \max \\ Ax + \omega &= b \\ x, \omega &\geq 0; \end{aligned} \qquad \text{Dual:} \quad \begin{aligned} y^T b &\longmapsto \min \\ A^T y - z &= c \\ y, z &\geq 0 \end{aligned}$$

where $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are given and we have also introduced the nonnegative ω and z slack variables to achieve equality conditions in the constraint system. This pair of problems is termed the *primal–dual pair in standard form.* If we perturb the objective function with the help of the logarithm function and a parameter $t > 0$, then the connected *log-barrier problem* is

$$\begin{aligned} c^T x + t \sum_{j=1}^{n} \log x_j + t \sum_{i=1}^{m} \log \omega_i &\longmapsto \max \\ Ax + \omega &= b \\ x, \omega &> 0 \end{aligned}$$

which is a one-parameter family of problems.

If the perturbed objectives have a unique optimum for each parameter, then the optimal solutions form a parametrized curve called the *central path.* As the parameter shrinks to zero, the perturbed objective approaches to the original one. Thus the central path is expected to terminate in the optimal solution of the original linear program.

To be able to study the central path in a geometrical point of view, we introduce the *path* and its *generator* in an abstract setting. (For precise details, consult the corresponding definitions of the chapter.)

The limit conditions (2.1) and (2.2) on the path generator ensure that the induced central path is continuous (see Lemma 2.2), and for special generators we have the monotonicity property along the central path as well (see Lemma 2.3).

The main results of CHAPTER 2 are presented in two theorems. The first one gives a sufficient condition for a concave program to have optimal feasible solution (Theorem 2.5).

THEOREM. *Assume that X and Y are Euclidean spaces, $D \subseteq X$ is a nonempty, open, convex set, $A \in \mathcal{L}(X, Y)$, and $b \in Y$. If a concave function $f\colon D \to \mathbb{R}$ satisfies the limit conditions*

$$\lim_{x \to x_0} f(x) = -\infty \qquad \lim_{\alpha \to +\infty} f(x + \alpha r) = -\infty$$

for all $x_0 \in \partial D$ and for all $r \in \operatorname{rec}(D)$, and the concave program

$$\begin{gathered} f(x) \longmapsto \max \\ Ax = b \end{gathered}$$

has a feasible solution, then it has a feasible optimal solution, as well.

The limit properties involved guarantee that the "large" values of the objective are allocated in a compact subset of the domain. The proof of this statement is based on the *open* sublevel sets of the objective function, and their recession cones also play a key role.

Using the concept of *standard projections* (for precise details, consult the corresponding definition of the chapter), the second main result of this chapter is the extension of the log-barrier problem (Theorem 2.6).

THEOREM. *Let $D \subseteq \mathbb{R}_+^{n+m}$ be a nonempty, open, convex set, $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, and $b \in \mathbb{R}^m$, furthermore $c \in \mathbb{R}^n$. Assume that $g\colon D \to \mathbb{R}$ is a continuously differentiable strictly concave function such that $\partial_k g$ are nonnegative, $\pi_k \partial_k g$ are bounded, and the function*

$$(x, \omega) \longmapsto c^T x + t g(x, \omega)$$

satisfies the limit conditions in Theorem 2.5 for all $t > 0$. If the original primal–dual pair has a feasible solution in D, then the central path of the family of concave programs

$$\begin{gathered} c^T x + t g(x, \omega) \longmapsto \max \\ Ax + \omega = b \end{gathered}$$

is well-defined and continuous. Moreover, each limit point at $t = 0$ of the central path's graph is an optimal solution of the attached primal–dual pair.

The one-parameter family of the concave problems reduces to a linear program in standard form if we set $t = 0$. The previous results (Lemma 2.2, Lemma 2.4, Theorem 2.5) ensure that its central path exists and it is continuous. Moreover, each limit point of its graph represents an optimal solution of the original primal–dual pair.

One advantage of our result is that we do not need second order tests to check optimality. Thus, simple *continuous* (instead of *twice*) differentiability assumption is enough on the objective. Therefore, Lagrange multipliers and the Complementary Slackness Theorem (see Theorem 2.1) from linear programming are sufficient for the proof.

As a direct application of the main results we can formulate the statement about the central path and the optimality of the classical log-barrier problem (Corollary 2.7).

COROLLARY. *If the original primal–dual pair has a positive feasible solution, then the central path of the log-barrier problem exists and continuous. Moreover, its each limit point at zero is a feasible optimal solution of the original primal–dual pair.*

The only thing that needs to be checked for this statement is that the log-barrier problem fulfills all the conditions of Theorem 2.6. This can be done with some simple calculations.

At the end of the chapter, we discuss that from the boundedness of the functions $\pi_k \partial_k g$ we can explain the usage of the logarithm as the perturbation function in the objective function of the classical barrier problem.

Consider now linear programs given in the general form

$$\begin{gathered} \langle c, x \rangle \longmapsto \max \\ x \in P \end{gathered}$$

where $P \subseteq \mathbb{R}^n$ is a polyhedron and $c \in \mathbb{R}^n$ is a given vector. This form allows us to focus on the geometric interpretation of linear programs. The main result of CHAPTER 3 (Theorem 3.3) gives a geometrical optimality condition for linear programs based on the well-known *graphical method.* This dimension-free characteristic property is that the coefficient vector of the objective has to make an obtuse angle with those half-lines that completely lie in the feasible set.

THEOREM. *The linear program in the general form has an optimal solution if and only if it is feasible and*

$$c \in \mathcal{N}_{\mathrm{rec}(P)}(0).$$

This statement might be well-known in the literature, however, our proof is based on the tools of Convex Analysis and Convex Geometry combined with induction. Let us note that it also suggests an algorithm that may differ from the usual simplex method since it moves on facets (and not between vertices).

Considering the fact that we can describe the polyhedron P as a mixed system of linear equalities and inequalities, for applications of the main result, we shall use the primal problem in *canonical form* while the dual problem in *standard form*:

$$\text{Primal:}\quad \begin{array}{rl} \langle c,x\rangle & \longmapsto \max \\ Ax & = b \\ x & \geq 0; \end{array} \qquad\qquad \text{Dual:}\quad \begin{array}{rl} \langle y,b\rangle & \longmapsto \min \\ A^T y & \geq c \end{array}$$

where $A \in \mathbb{R}^{m\times n}$ is a given matrix, further, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are given vectors. With the help of our main result, we can immediately give the optimality condition for these problems (Corollary 3.4, Corollary 3.5).

COROLLARY. *A maximum problem in canonical form has an optimal solution if and only if it is feasible and* $\langle c,r\rangle \leq 0$ *for all* $r \in \mathrm{Ker}(A) \cap \mathbb{R}^n_+$.

COROLLARY. *A minimum problem in standard form has an optimal solution if and only if it is feasible and* $\langle b,q\rangle \geq 0$ *whenever* $A^T q \geq 0$.

Their proof is a simple calculation based on Lemma 1.6 and Lemma 3.1. These corollaries may be applied to quickly check whether a linear program has an optimal solution or not.

The strong duality theorem (see Corollary 3.6) is one of the most important cornerstone of linear programming. With the basic separation theorem (Lemma 1.1) and our main result we can give an alternative proof avoiding the simplex method. Furthermore, if the primal–dual problems have the above form, then we can also deduce Farkas' lemma, which is well-known to be equivalent to the strong duality theorem (Corollary 3.7).

COROLLARY. *If* $A \in \mathbb{R}^{m\times n}$ *and* $b \in \mathbb{R}^m$ *then*

(1) either the system $Ax = b$ *and* $x \geq 0$ *has a solution;*
(2) or the system $A^T y \geq 0$ *and* $b^T y < 0$ *has a solution.*

Let us mention that their proof allow a geometric approach to duality theory in linear programming besides the algorithmic ones.

The aim of CHAPTER 4 is to present an algorithm that decides the solvability of planar feasible linear programs. This method is motivated by the main result in CHAPTER 3 and the fact that in everyday life problems sometimes only the solvability issue of a linear program counts in a practical application and the solution itself (if it exists) does not. Let us emphasize that this algorithm is different from the one that was suggested by the proof of Theorem 3.3.

For the implementation of the method we need to work with the planar linear programming problem in the form

$$\begin{aligned} c_1x_1 + c_2x_2 &\longmapsto \max \\ a_{i1}x_1 + a_{i2}x_2 &\leq b_i \end{aligned}$$

where $c_j, a_{ij}, b_i \in \mathbb{R}$ are given for all $j \in \{1,2\}$ and $i \in \{1,\dots,m\}$ indices. Our method for testing the solvability of feasible linear programs given in this form is the following.

Firstly, we reduce the list of normal vectors (a_{i1}, a_{i2}) with an equivalence classification such that we norm them to the unit circle of $\mathbb{R}^2$ and collect one representative of each class into the reduced point list R. Then, from R we collect all the opposing pair of points (if there is any) into a new list N. The main task is to determine the list E containing the extremal vectors that generate the normal cone of the feasible set. Depending on the lists R and N, this part of the algorithm has several stopping points when it can achieve its output value (the list E). Finally, the main theorem in this chapter (Theorem 4.1) decides the solvability of the given planar linear programming problem.

THEOREM. *The previously presented method is mathematically correct and it stops in finitely many steps. Moreover, if the problem is feasible and at the end of Algorithm 4.3*

(1) E is empty, then there exists an optimal solution;
(2) E is nonempty and $c \in \operatorname{cone}(E)$, then there exists an optimal solution, otherwise not.

In particular, if $c \notin \operatorname{cone}(E)$, then there is no optimal solution.

The proof of this statement, which is also a validation of our method, can be done in a geometrical way. It is a separation of cases based on the number of elements in the list N and the separation property by lines chosen in a special way. However, in every case, the solvability is ensured by Theorem 3.3.

One of the disadvantages of this algorithm is that it does not ensure the feasibility of the given linear program. Thus, this method might be more useful in general for determining the non-solvability of the linear program.

Considering this method in higher dimensions, it might be possible to formulate a similar one, however, some further problems might occur. We cannot apply the concept of “maximum angle” of two points as we did it in the planar case. Moreover, our method heavily depends on the fact that every convex cone in $\mathbb{R}^2$ (if it is not a half-plane) can be uniquely generated by at most 2 extremal vectors and this might reduce the required calculation steps. But this fact is not ensured even in the 3-dimensional case since the number of extremals can be very high.

Összefoglalás

Az értekezés motivációja egy másik algoritmuscsalád, nevezetesen a belső-pontos-módszerek közé tartozó logaritmikus sorompó probléma újratárgyalása. A belső pontos módszerekről igazolták, hogy van köztük polinomiális időben futó algoritmus, vagyis a lineáris programozási feladatok polinomiális komplexitásúak. A lineáris programozási problémák speciális geometriai tulajdonságai továbbá lehetővé teszik geometriai feltételek és új típusú módszerek kifejlesztését. Az értekezés célja a lineáris programozási feladatoknak a konvex analízis és a konvex geometria eszközeivel való megközelítése úgy, hogy elkerüljük a szimplex módszer használatát. Megemlítjük, hogy az értekezés egyes fejezetei rendre az [**5**], [**6**] cikkekben közölt eredményekre támaszkodnak.

Az értekezés első fejezetében összegyűjtjük a konvex analízis és a konvex geometria mindazon eszközeit, amelyeket vagy a legtöbb fejezet során vagy az egész értekezésben általánosan használunk, mint például a konvex és konkáv függvények definícióját; a Bernstein–Doetch tételt; illetve a halmazok konvex és kúp burkának definícióját.

Az elválasztási tételről általánosan elmondható, hogy jelentős szerepet játszik az optimalizálási problémakörben. Az euklideszi változata az alábbi módon fogalmazható meg.

LEMMA. *Euklideszi terekben bármely nem üres, zárt és konvex halmaz hipersíkkal szigorúan elválasztható bármely, a halmazhoz nem tartozó ponttól.*

Erre a változatra számos bizonyítás is ismert, Barvinok azonban egy alternatív megközelítést javasol kitűzött feladatként, amelyre megoldást adunk ebben a fejezetben.

Jelölje $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ az összes $\mathbb{R}^n$-ből $\mathbb{R}^m$-be ható lineáris leképezések terét. Az ilyen lineáris transzformációk még a legegyszerűbb esetekben is elronthatják a (konvex) halmazok zártságát. Azonban, ha a halmaz egy poliéder, akkor a lineáris transzformáltja is poliéder marad. Ennek a ténynek a bizonyítása meglehetősen bonyolult, de szerencsére elkerülhetjük ezt a kellemetlenséget a következő egyszerű megfigyeléssel.

LEMMA. *Vektorterekben bármely végesen generált kúp a függetlenül generált részkúpok egyesítése. Speciálisan, egy véges dimenziós tér bármely végesen generált kúpja zárt.*

Ennek az állításnak a bizonyítása a Carathéodory-tétel kúpokról szóló változatára támaszkodik.

A feltételes optimalizálási feladatokat és a hozzájuk kapcsolódó alapfogalmakat általánosan az alábbi módon írhatjuk fel.

DEFINÍCIÓ. *Legyenek X és Y nemüres halmazok, továbbá $D \subseteq X$ és $b \in Y$. Tekintsük az*

$$f(x) \longmapsto \max$$
$$g(x) = b$$

feltételes optimalizálási feladatot, ahol az $f\colon D \to \mathbb{R}$ célfüggvény *és a $g\colon X \to Y$* feltétel *adottak. Azt mondjuk, hogy $x \in X$ egy* megengedett megoldás*, ha $x \in D$ és $g(x) = b$ teljesül. Továbbá, az összes megengedett megoldások halmazát a probléma* megengedett halmaz*ának nevezzük. Az $x^* \in X$ elemet megengedett* optimális megoldás*nak hívjuk, ha megengedett halmazbeli és $f(x) \leq f(x^*)$ teljesül minden $x \in X$ megengedett megoldás esetén.*

A lineáris programozási feladatokat az előbbi általános feladatok speciális eseteként vezetjük be: a problémának mind a célfüggvényét, mind a feltételi rendszerét lineáris kifejezésekkel adjuk meg. Az előző definícióban legyen $X = \mathbb{R}^n$ és $Y = \mathbb{R}^m$. Ekkor a

$$\langle c, x\rangle \longmapsto \max$$
$$x \in P$$

feltételes szélsőérték számítási feladatot *lineáris programozási feladat*nak hívjuk, ahol $c \in \mathbb{R}^n$ egy adott vektor, $P \subseteq \mathbb{R}^n$ pedig egy *poliéder*, azaz véges sok féltér metszete. A többi kapcsolódó fogalom analóg módon vezethető be az előző definíció alapján.

A konvex halmazok recessziós irányai döntő szerepet játszanak az értekezésben kitűzött céljaink megvalósításában.

DEFINÍCIÓ. *Legyen X vektortér, D egy nemüres részhalmaza X-nek, és legyen $x \in D$ rögzített. Azt mondjuk, hogy $r \in X$ a D halmaz x pontbeli* recessziós iránya*, ha minden $\alpha \geq 0$ esetén $x + \alpha r \in D$ teljesül.*

Rockafellar azt az esetet vizsgálta, amikor D egy *zárt*, konvex halmaz. Kiderült, hogy D recessziós irányai számos fontos tulajdonsággal rendelkeznek. A recessziós irányok függetlenek az $x \in D$ megválasztásától és a recessziós irányok létezése jellemzi D nemkorlátosságát. Következésképpen D recessziós irányainak halmazát jelölhetjük egyszerűen $\operatorname{rec}(D)$ módon. Továbbá, a recessziós irányok kúpot alkotnak: zártak az összeadásra és a nemnegatív számmal való szorzásra nézve. Ezért a $\operatorname{rec}(D)$ halmazt D *recessziós kúp*jának nevezzük.

A logaritmikus sorompó probléma célfüggvényének értelmezési tartománya azonban megköveteli, hogy *nyílt*, konvex halmazokat használjunk. Szerencsére az előbb említett tulajdonságok ebben az esetben is igazak maradnak, amit az alábbi lemmában fogalalunk össze. Az alapponttól való függetlenség miatt a D halmaz recessziós kúpját továbbra is $\operatorname{rec}(D)$ módon jelölhetjük.

LEMMA. *Ha D egy nemüres, nyílt, konvex részhalmaza egy X Euklideszi térnek, továbbá $r \in X$, akkor az alábbi állítások ekvivalensek:*

(1) Létezik $x \in D$ úgy, hogy minden $\alpha \geq 0$ esetén $x + \alpha r \in D$ teljesül;
(2) Minden $x \in D$ és minden $\alpha \geq 0$ esetén $x + \alpha r \in D$ teljesül.

Továbbá, D pontosan akkor korlátos, ha $\operatorname{rec}(D)$ *triviális.*

A hasonló tulajdonságok ellenére azonban a bizonyítás során alternatív megközelítést kell követnünk.

Megemlítjük végül a recessziós kúpok következő kalkulus tulajdonságát is, amely mind a nyílt, mind a zárt esetben is érvényes, valamint a bizonyításukban lévő érvelés is nagyon hasonló.

LEMMA. *Legyen X Euklideszi tér. Ha $A, B \subseteq X$ olyan nyílt/zárt, konvex halmazok, amelyek metszete nemüres, akkor*

$$\operatorname{rec}(A \cap B) = \operatorname{rec}(A) \cap \operatorname{rec}(B).$$

A második fejezet célja a klasszikus logaritmikus sorompó probléma kiterjesztése úgy, hogy az érvelések során kerüljük a szimplex módszer használatát, emellett korrigáljuk Vanderbei hibás érvelését: A Heine–Borel tételt $\mathbb{R}^n$ egy nyílt részhalmaza által indukált relatív topológiában alkalmazta. Ehhez tekintsük az alábbi lineáris programozási feladatpárt:

$$\text{Primál:}\quad \begin{aligned} c^T x &\longmapsto \max \\ Ax + \omega &= b \\ x, \omega &\geq 0; \end{aligned} \qquad\qquad \text{Duál:}\quad \begin{aligned} y^T b &\longmapsto \min \\ A^T y - z &= c \\ y, z &\geq 0, \end{aligned}$$

ahol $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, $c \in \mathbb{R}^n$ és $b \in \mathbb{R}^m$ adottak, valamint a feltételrendszerbe már bevezettük az ω és z nemnegatív segédváltozókat annak érdekében, hogy egyenlőségi feltételeket kapjunk. Ezt a problémapárt *standard alakban adott primál–duál feladatpár*nak nevezzük. Ha a primál célfüggvényt perturbáljuk a változók logaritmusával, melyet egy $t > 0$ paraméterrel is kiegészítünk, akkor a kapcsolódó *logaritmikus sorompó probléma* az alábbi egyparaméteres problémacsalád:

$$\begin{aligned} c^T x + t \sum_{j=1}^{n} \log x_j + t \sum_{i=1}^{m} \log \omega_i &\longmapsto \max \\ Ax + \omega &= b \\ x, \omega &> 0. \end{aligned}$$

Ha az új céfüggvénynek minden t paraméter esetén egyértelmű optimuma van, akkor az optimális megoldások egy paraméteres görbét alkotnak, amelyet *centrális út*nak nevezünk. Ahogy a t paraméter értéke tart nullához, a sorompó probléma célfüggvénye megközelíti az eredetit, így a centrális út várhatóan az eredeti lineáris programozási feladat optimális megoldásához tart.

Annak érdekében, hogy a centrális utat geometriai szempontból vizsgálhassuk, bevezetjük azt egy absztrakt környezetben annak generátorfüggvényével együtt.

DEFINÍCIÓ. *Legyenek T és B nemüres halmazok. Azt mondjuk, hogy az $F\colon T\times B \to \mathbb{R}$ leképezés teljesíti a* paraméteres globális maximum tulajdonság*ot, ha minden $t \in T$ esetén egyértelműen létezik egy $h(t) \in B$ úgy, hogy*

$$F\big(t, h(t)\big) > F(t, x)$$

teljesül minden $x \neq h(t)$ mellett. Ekkor, $h\colon T \to B$ valóban függvény. A h függvényt ekkor centrális út*nak, az F leképezést pedig a centrális út* generátor*ának nevezzük.*

Ha T metrikus tér és B egy E euklideszi tér nemüres részhalmaza, akkor az $F\colon T \times B \to \mathbb{R}$ generátor esetén tekintsük az alábbi határérték feltételeket: Minden $t_0 \in T$ és minden $x_0 \in \overline{B} \setminus B$ esetén

$$\lim_{\substack{t\to t_0\\ x\to x_0}} F(t, x) = -\infty.$$

Valamint minden $t_0 \in T$ esetén

$$\lim_{\substack{t\to t_0\\ \|x\|\to +\infty}} F(t, x) = -\infty.$$

Ezek a feltételek garantálják a sorompó probléma centrális útjának folytonosságát. Emellett, speciális alakú generátor függvények esetén a centrális út mentén monotonitás is biztosítható.

A második fejezet fő eredményeit két tételben foglaljuk össze. Az első elegendő feltételt ad ahhoz, hogy egy konkáv célfüggvénnyel rendelkező problémának létezzen megengedett optimális megoldása.

TÉTEL. *Tegyük fel, hogy X és Y Euklideszi terek, $D \subseteq X$ egy nemüres, nyílt, konvex halmaz, $A \in \mathcal{L}(X, Y)$, és $b \in Y$. Ha egy $f\colon D \to \mathbb{R}$ konkáv függvény teljesíti a*

$$\lim_{x\to x_0} f(x) = -\infty \qquad \lim_{\alpha\to +\infty} f(x + \alpha r) = -\infty$$

határérték feltételeket minden $x_0 \in \partial D$ és minden $r \in \operatorname{rec}(D)$ esetén, valamint az

$$\begin{aligned} f(x) &\longmapsto \max \\ Ax &= b \end{aligned}$$

konkáv programozási problémának van megengedett megoldása, akkor létezik megengedett optimális megoldása is.

A feltételek között szereplő határérték tulajdonságok garantálják, hogy a célfüggvény „nagy” értékei az értelmezési tartomány egy kompakt részhalmazában csoportosulnak. Az állítás bizonyítása a célfüggvény *nyílt* szinthalmazain alapul, és bizonyos ponton azok recessziós kúpjai is kulcsszerepet kapnak.

A második fő eredmény a logaritmikus sorompó probléma általánosítása. Ennek megfogalmazása előtt még bevezetjük a standard projekció fogalmát is, amely szerepet kap a tétel feltételei között.

DEFINÍCIÓ. *Legyen X egy euklideszi tér és tekintsük a standard bázisát. Ha $1 \leq k \leq \dim(X)$, akkor a k-adik* standard projekció *az a $\pi_k \colon X \to \mathbb{R}$ függvény, amely az x_k standard koordinátát rendeli hozzá az összes $x \in X$ elemhez.*

TÉTEL. *Legyen $D \subseteq \mathbb{R}_+^{n+m}$ egy nemüres, nyílt, konvex halmaz, $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, és $b \in \mathbb{R}^m$, továbbá $c \in \mathbb{R}^n$. Tegyük fel, hogy $g \colon D \to \mathbb{R}$ egy olyan folytonosan differendciálható, szigorúan konkáv függvény, hogy $\partial_k g$ nemnegatívak, $\pi_k \partial_k g$ korlátosak, és az*

$$(x, \omega) \longmapsto c^T x + tg(x, \omega)$$

leképezés teljesíti az előző tételbeli határérték feltételeket minden $t > 0$ esetén. Ha az eredeti primál–duál párnak létezik D-beli megengedett megoldása, akkor a

$$\begin{gathered} c^T x + tg(x, \omega) \longmapsto \max \\ Ax + \omega = b \end{gathered}$$

konkáv programozási feladatcsaládhoz tartozó centrális útja létezik és folytonos. Továbbá, a centrális út gráfjának minden $t = 0$ paraméterhez tartozó torlódási pontja a kapcsolódó primál–duál feladatpár egy optimális megoldása.

Megfigyelhető, hogy az egyparaméteres problémacsalád a $t = 0$ választás mellett az eredeti standard formában adott lineáris programozási feladatra redukálódik. Az előző eredmények biztosítják a centrális út létezését és annak folytonosságát, továbbá, a centrális út gráfjának minden torlódási pontja az eredeti primál–duál feladatpár egy optimális megoldása.

Az általánosításunk egyik jelentős előnye, hogy nincs szükségünk másodrendű tesztre az optimalitás ellenőrzéséhez. A célfüggvényre ezért elegendő az egyszerű *folytonos* differenciálhatósági feltétel a *kétszeres* differenciálhatóság helyett. Így a bizonyításhoz elegendő csupán a Lagrange-féle multiplikátor-elvet és a lineáris programozásban jól ismert Komplementaritási tételt alkalmaznunk.

A fő eredmények közvetlen alkalmazásaként megfogalmazhatjuk a klasszikus logaritmikus sorompó probléma centrális útjáról és optimalitásáról szóló állítást.

KÖVETKEZMÉNY. *Ha az eredeti primál–duál feladatpár mindegyikének van pozitív megengedett megldása, akkor a kapcsolódó sorompó-probléma centrális útja létezik és folytonos. Továbbá, a centrális út gráfjának bármely nullabeli torlódási pontja az eredeti problémapár megengedett optimális megoldása.*

Az állítás bizonyításához mindössze annyit kell ellenőrizni, hogy a logaritmikus sorompó probléma célfüggvénye teljesíti a második fő eredményünkben szabott feltételeket. Ez néhány egyszerű számolással megtehető.

A fejezet zárásaként végül kitérünk arra, hogy a $\pi_k \partial_k g$ függvények korlátosságából kiindulva magyarázható a logaritmus, mint perturbáló függvény használata a klasszikus sorompó probléma célfüggvényében.

Tekintsük most a lineáris programozási feladatok korábban már említett általános alakját:

$$\begin{aligned}\langle c, x\rangle &\longmapsto \max\\ x &\in P.\end{aligned}$$

Ez a fajta felírás lehetővé teszi a lineáris programozási feladatok geometriai megközelítését. A harmadik fejezet fő eredménye egy geometriai optimalitási feltételt ad lineáris programozási feladatokhoz a jól ismert *grafikus módszer* alapján. Ez a dimenziófüggetlen karakterisztikus tulajdonság nem más, mint hogy a célfüggvény együtthatóvektorának tompaszöget kell bezárnia azokkal a félegyenesekkel, amelyek teljes egészében a megengedett halmazon belül helyezkednek el. Ezt a tulajdonságot $\mathrm{rec}(P)$ *normálkúp*ja segítségével írhatjuk le pontosan.

DEFINÍCIÓ. *Ha D egy X vektortér konvex részhalmaza és $p \in D$, akkor az*

$$\mathcal{N}_D(p) := \{y \in X \mid \langle y, x-p\rangle \leq 0,\ x \in D\}$$

halmazt a D halmaz p pontbeli normál kúp*jának nevezzük.*

TÉTEL. *A fenti lineáris programozási feladatnak pontosan akkor létezik optimális megoldása, ha megengedett és*

$$c \in \mathcal{N}_{\mathrm{rec}(P)}(0).$$

Ez az állítás jól ismert lehet a szakirodalomban, azonban a bizonyítás a konvex analízis és a konvex geometria eszközeire épül, amit teljes indukciós lépésekkel kombinálunk. Fontos megjegyezni, hogy emiatt a bizonyítás egy algoritmust is javasol, amely eltérhet a szimplex módszertől, hiszen oldalak mentén mozog, nem pedig csúcsok között, ahogyan a szimplex módszer teszi.

Figyelembe véve azt a tényt, hogy a P poliédert lineáris egyenlőségek és egyenlőtlenségek vegyes rendszereként is felírhatjuk, a fő eredmény alkalmazásaihoz a primál problémát tekintsük *kanonikus*, míg a duális problémát *standard* alakban:

$$\text{Primál:}\quad \begin{aligned}\langle c, x\rangle &\longmapsto \max\\ Ax &= b\\ x &\geq 0;\end{aligned} \qquad\qquad \text{Duál:}\quad \begin{aligned}\langle y, b\rangle &\longmapsto \min\\ A^T y &\geq c,\end{aligned}$$

ahol $A \in \mathbb{R}^{m\times n}$ egy $m \times n$-es mátrix, továbbá $c \in \mathbb{R}^n$ és $b \in \mathbb{R}^m$ adott vektorok. Az ilyen módon adott feladatok optimalitási feltételét azonnal megadhatjuk a fő eredményünk segítségével az alábbi módon.

KÖVETKEZMÉNY. *Egy kanonikus formában adott maximum-feladatnak pontosan akkor létezik optimális megoldása, ha megengedett és $\langle c, r\rangle \leq 0$ teljesül minden $r \in \mathrm{Ker}(A) \cap \mathbb{R}^n_+$ esetén.*

KÖVETKEZMÉNY. *Egy standard formában adott minimum-feladatnak pontosan akkor létezik optimális megoldása, ha megengedett és $\langle b, q\rangle \geq 0$ teljesül, valahányszor $A^T q \geq 0$.*

Mindkét állítás egy egyszerű számolás eredménye, azonban ezek a következmények jól alkalmazhatók annak gyors ellenőrzésére, hogy egy lineáris programozási feladatnak van-e optimális megoldása vagy sem.

Az erős dualitási tételt a lineáris programozás egyik alapkövének tekinthető.

KÖVETKEZMÉNY. *Ha egy primál feladatnak létezik optimális megoldása, akkor a duál feladatnak is van optimális megoldása.*

Az elválasztási tétel és fő eredményünk segítségével alternatív bizonyítást adhatunk a szimplex módszertől függetlenül. Továbbá, ha a primál-duális problémák a fenti alakúak, akkor levezethetjük Farkas-lemmát is, amelyről köztudott, hogy ekvivalens az erős dualitás tételével.

KÖVETKEZMÉNY. *Ha* $A \in \mathbb{R}^{m\times n}$ *és* $b \in \mathbb{R}^m$*, akkor*

(1) vagy az $Ax = b$ *és* $x \geq 0$ *rendszernek van megoldása;*
(2) vagy az $A^T y \geq 0$ *és* $b^T y < 0$ *rendszernek van megoldása.*

Megjegyezzük továbbá, hogy ezek bizonyítása lehetővé teszi a dualitáselmélet geometriai megközelítését is az algoritmikusak mellett.

A negyedik fejezet célja egy olyan algoritmus bemutatása, amely eldönti a síkbeli lineáris programozási feladtok megoldhatóságát. Ezt a módszert a harmadik fejezet fő eredménye motiválja, valamint az a tény, hogy a hétköznapi életben felmerülő problémák során néha csupán az adott lineáris programozási feladat megoldhatósági kérdése számít, és maga az optimális megoldás (ha létezik) nem. Hangsúlyozzuk, hogy ez az algoritmus különbözik attól, amelyet az előző fejezet fő tételének bizonyítása javasolt.

A módszer implementálásához a síkbeli lineáris programozási problémákat a következő formában kell megadnunk:

$$c_1x_1 + c_2x_2 \longmapsto \max$$
$$a_{i1}x_1 + a_{i2}x_2 \leq b_i,$$

ahol $c_j, a_{ij}, b_i \in \mathbb{R}$ adott minden $j \in \{1,2\}$ és $i \in \{1,\dots,m\}$ indexre. Az ilyen alakban adott megengedett lineáris programozási feladatok megoldhatóságának tesztelésére szolgáló módszerünk alapgondolata a következő.

Először a megengedett halmazt meghatározó (a_{i1}, a_{i2}) normálvektorokon osztályozást végzünk úgy, hogy ezeket a vektorokat $\mathbb{R}^2$ egységkörére normáljuk, majd minden osztályból egy reprezentatív elemet tartunk meg az R-rel jelölt redukált pontlistában. Ezután R-ből összegyűjtjük az ellenlakó pontpárokat (ha vannak) egy új N listába. Az algoritmus fő része az E lista meghatározása, amely a megengedett halmaz normál kúpját generáló extremálisokat tartalmazza. Az algoritmus itt már több ponton megállhat az R és az N listáktól függően. Végezetül, a fejezet fő tétele dönti el az adott síkbeli lineáris programozási probléma megoldhatóságát.

TÉTEL. *Az előbb bemutatott módszer matematikailag korrekt és véges lépésben megáll. Továbbá, ha a probléma megengedett és az algoritmus végén*

(1) *E üres, akkor létezik optimális megoldás;*
(2) *E nemüres és $c \in \operatorname{cone}(E)$, akkor létezik optimális megoldás, egyébként nem.*

Speciálisan, ha $c \notin \operatorname{cone}(E)$, akkor a feladatnak nincs optimális megoldása.

Ez az állítás geometriailag bizonyítható, amely egyben a módszerünk validálása is, és az N lista elemszáma szerinti esetszétválasztáson, valamint speciálisan választott egyenesek általi szeparáláson alapul. Azonban minden esetben a megoldhatóságot a harmadik fejezet fő eredménye biztosítja.

Ennek az algoritmusnak az egyik hátránya, hogy nem biztosítja az adott lineáris programozási feladat megengedettségét. Így általánosabban ez a módszer inkább a lineáris programozási feladatok nemmegoldatóságának ellenőrzésére alkalmasabb!

A módszer magasabb dimenziókba való átültetése elképzelhető lehet hasonló alapokra támaszkodva, azonban ezzel egyidejűleg több probléma is felmerülhet. Nem alkalmazhatjuk két pont „maximális szögének” koncepcióját úgy, ahogyan azt a 2-dimenziós esetben tettük. Továbbá, ez a módszer nagymértékben függ attól a ténytől, hogy a síkban minden konvex kúpot (ha az nem egy félsík) legfeljebb 2 extremális vektor egyértelműen generál, ami csökkentheti az algoritmus számolási igényét. Viszont, az extremálisok száma tetszőlegesen nagy lehet, már a 3-dimenziós esetben is, így a számolási igény sem csökkenthető!

Bibliography

[1] Barvinok, A., *A course in convexity*, Graduate Studies in Mathematics, vol. 54, American Mathematical Society, Providence, R.I., 2002.

[2] Bayer, D., Lagarias, J., *The nonlinear geometry of linear programming: affine and projective scaling trajectories*, Transactions of the AMS **314** (1989), 499–525.

[3] Bayer, D., Lagarias, J., *The nonlinear geometry of linear programming: Legendre transform coordinates and central trajectories*, Transactions of the AMS **314** (1989), 527–581.

[4] Bernstein, F., Doetsch, G., *Zur Theorie der konvexen Funktionen*, Math. Ann. **76** (1915), no. 4, 514–526.

[5] Bessenyei, M., Tóth, N., *A Convex Analysis View of the Barrier Problem*, Journal of Convex Analysis **29** (2022), no. 3, 827–836.

[6] Bessenyei, M., Tóth, N., *An optimality condition for linear programs*, Publicationes Mathematicae Debrecen **108** (2026).

[7] Borwein, J. M., Lewis, A. S., *Convex analysis and nonlinear optimization – Theory and examples*, CMS Books in Mathematics, vol. 3, 2nd ed., Springer, New York, 2006.

[8] Carathéodory, C., *Über den Variabilitätsbereich der Fourierschen Konstanten von positiven harmonischen Funktionen*, Rend. Circ. Mat. Palermo **32** (1911), 193–217.

[9] Dantzig, G. B., *Application of the simplex method to a transportation problem*, Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13 (1951), 359–373.

[10] Dantzig, G. B., *Linear programming and extensions*, Princeton University Press, Princeton, N.J., 1963.

[11] Farkas, J., *Theorie der einfachen Ungleichungen*, J. für Math. **124** (1901), 1–27.

[12] Fiacco, A. V., McCormick, G. P., *Nonlinear programming: Sequential unconstrained minimization techniques*, John Wiley and Sons, Inc., New York-London-Sydney, 1968.

[13] Gale, D., *The theory of linear economic models*, McGraw-Hill Book Co., Inc., New York-Toronto-London, 1960.

[14] Gordan, P., *Über die Auflösung linearer Gleichungen mit reellen Coefficienten*, Math. Ann. **6** (1873), 23–28.

[15] Huard, P., *Resolution of mathematical programming with nonlinear constraints by the method of centers*, in J. Abadie, ed., Nonlinear Programming, North-Holland, Amsterdam, (1967), pp. 209–219.

[16] Kantorovich, L., *Mathematical methods in the organization and planning of production*, Management Science **6** (1960), 550–559.

[17] Karmarkar, N., *A new polynomial time algorithm for linear programming*, Combinatorica **4** (1984), 373–395.

[18] Khachiyan, L. G., *A polynomial algorithm in linear programming*, Dokl. Akad. Nauk SSSR, vol. 244, no. 5 (1979), 1093–1096.

[19] Klee, V., Minty, G. J., *How good is the simplex algorithm?*, Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969, 159–175, 1972.

[20] Komornik, V., *A simple proof of Farkas' lemma*, Amer. Math. Monthly **105** (1998), no. 10, 949–950.

[21] Megiddo, N., *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, pp. 131–158, Springer-Verlag, New York, 1989.

[22] Minkowski, H., *Geometrie der Zahlen*, Bibliotheca Mathematica Teubneriana, Band 40, Johnson Reprint Corp., New York-London, 1968.

[23] Motzkin, Th. S., *Beiträge zur Theorie der linearen Ungleichungen*, Ph.D. Diss., Basel, 1936.

[24] Rockafellar, R., *Convex Analysis*, Princeton University Press, Princeton, N.J., 1970.

[25] Roos, C., Terlaky, T., Vial, J. P., *Interior point methods for linear optimization*, Springer, New York, 2006.

[26] Schrijver, A., *Theory of linear and integer programming*, Wiley-Interscience Series in Discrete Mathematics, A Wiley-Interscience Publication, John Wiley & Sons, Ltd., Chichester, 1986.

[27] Stiemke, E., *Über positive Lösungen homogener linearer Gleichungen*, Math. Ann. **76** (1915), 340–342.

[28] Tucker, A. W., *Dual systems of homogeneous linear relations*, Linear inequalities and related systems, Annals of Mathematics Studies, no. 38, 3–18, Princeton University Press, Princeton, N.J., 1956.

[29] Vanderbei, R. J., *Linear programming – Foundations and extensions*, International Series in Operations Research & Management Science, vol. 196, 4th ed., Springer, New York, 2014.

[30] Ville, J., *Sur la théorie général des jeux ou intervient l'habileté des jouers*, in E. Borel, ed., 'Traité du calcul des probabilités et de ses applications', Paris, Gauthier-Villars, 1938.

[31] Weyl, H., *Elementare Theorie der konvexen Polyeder*, Comment. Math. Helv., **7** (1934), no. 1, 290–306.

[32] Wright, S. J., *Primal-dual interior-point methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

Appendices

```
> ### Necessary Maple package for plotting:

  with(plots);
```

```
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,            (1)
   conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
   display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot,
   implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot,
   listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple,
   odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d,
   polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions,
   setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]
```

```
> ListReduction:=proc(A)
   local L,L_red,i,j,la;

   ### Input: List of original normal vectors generated by the rows
  of A

   L:=A;

   if numelems(L)=1 then
    return L;

   else

    L_red:=A;
    for i from 1 to numelems(L)-1 do
     for j from i+1 to numelems(L) do
      la:=solve({L[i][1]*la1=L[j][1],L[i][2]*la2=L[j][2]});
      if numelems(table(la))=1 then
       if rhs(la[1])>0 then
        L_red:=remove(is,L_red,L[j]);
       end if;
      elif numelems(table(la))=2 then
       if rhs(la[1])>0 and rhs(la[1])=rhs(la[2]) then
        L_red:=remove(is,L_red,L[j]);
       end if;
      end if;
     end do;
    end do;
    return L_red;

   end if;

   ### Output: Reduced point list L_red

  end proc;
```

```
ListReduction := proc(A)                                                                        (2)
   local L, L_red, i, j, la;
   L := A;
   if numelems(L) = 1 then
      return L
```

```
    else
        L_red := A;
        for i to numelems(L) − 1 do
            for j from i + 1 to numelems(L) do
                la := solve( {L[i][1] * la1 = L[j][1], L[i][2] * la2 = L[j][2]});
                if numelems(table(la)) = 1 then
                    if 0 < rhs(la[1]) then L_red := remove(is, L_red, L[j]) end if
                elif numelems(table(la)) = 2 then
                    if 0 < rhs(la[1]) and rhs(la[1]) = rhs(la[2]) then
                        L_red := remove(is, L_red, L[j])
                    end if
                end if
            end do
        end do;
        return L_red
    end if
end proc
```

```
> OpposingPairs:=proc(A)
   local L_red,L_neg,i,j,la;

   ### Calling the procedure ListReduction on A to achieve the
  input value L_red

   L_red:=ListReduction(A);

   L_neg:=[];

   if numelems(L_red)=1 then
    return L_red,L_neg;

   else

    for i from 1 to numelems(L_red)-1 do
     for j from i+1 to numelems(L_red) do
      la:=solve({L_red[i][1]*la1=L_red[j][1],L_red[i][2]*la2=L_red
  [j][2]});
      if numelems(table(la))=1 then
       if rhs(la[1])<0 then
        L_neg:=[op(L_neg),[L_red[i],L_red[j]]]
       end if;
      elif numelems(table(la))=2 then
       if rhs(la[1])<0 and rhs(la[1])=rhs(la[2]) then
        L_neg:=[op(L_neg),[L_red[i],L_red[j]]]
       end if;
      end if;
     end do;
    end do;
    return L_red,L_neg;

   end if;
```

```
 ### Output: Reduced point list L_red; list L_neg containing
opposing pair of points

end proc;
```

```
OpposingPairs := proc(A)                                                    (3)
   local L_red, L_neg, i, j, la;
   L_red := ListReduction(A);
   L_neg := [ ];
   if numelems(L_red) = 1 then
      return L_red, L_neg
   else
      for i to numelems(L_red) − 1 do
         for j from i + 1 to numelems(L_red) do
            la := solve({L_red[i][1] * la1 = L_red[j][1], L_red[i][2] * la2 = L_red[j][2
            ]});
            if numelems(table(la)) = 1 then
               if rhs(la[1]) < 0 then L_neg := [op(L_neg), [L_red[i], L_red[j]]]
               end if
            elif numelems(table(la)) = 2 then
               if rhs(la[1]) < 0 and rhs(la[1]) = rhs(la[2]) then
                  L_neg := [op(L_neg), [L_red[i], L_red[j]]]
               end if
            end if
         end do
      end do;
      return L_red, L_neg
   end if
end proc
```

```
> NormalCone:=proc(A)
 local L_red,L_neg,L_normdir,X,Y,i,j,line,X0,Y0,sign1,sign2,
sign_list,m,TrueFalse;

 ### Calling the procedures OpposingPairs on A to achieve the
input values L_red and L_neg

 L_red:=OpposingPairs(A)[1];
 L_neg:=OpposingPairs(A)[2];
 L_normdir:=[];

 if numelems(L_red)=1 then
  L_normdir:=L_red;

 else

  if numelems(L_neg)=0 then

   X:=[seq(L_red[i][1],i=1..numelems(L_red))];
   Y:=[seq(L_red[i][2],i=1..numelems(L_red))];
```

```
   for i from 1 to numelems(X) while numelems(L_normdir)<2 do
    line:=Y[i]*x-X[i]*y;
    X0:=subsop(i=NULL,X);
    Y0:=subsop(i=NULL,Y);
    sign1:=evalb(eval(eval(line,x=X0[1]),y=Y0[1])<=0);
    m:=0;
    for j from 2 to numelems(X0) while m=0 do
     sign2:=evalb(eval(eval(line,x=X0[j]),y=Y0[j])<=0);
     if sign1<>sign2 then
      m:=m+1;
     end if;
    end do;
    if m=0 then
     L_normdir:=[op(L_normdir),[X[i],Y[i]]];
    end if;
   end do;

  elif numelems(L_neg)=1 then
   if numelems(L_red)=2 then
    L_normdir:=L_red;
   elif numelems(L_red)>=3 then
    for i from 1 to numelems(L_neg[1]) do
     L_red:=remove(is,L_red,L_neg[1][i]);
    end do;

    X:=[seq(L_red[i][1],i=1..numelems(L_red))];
    Y:=[seq(L_red[i][2],i=1..numelems(L_red))];
    line:=L_neg[1][1][2]*x-L_neg[1][1][1]*y;
    sign_list:=[];
    for i from 1 to numelems(X) do
     sign_list:=[op(sign_list),evalb(eval(eval(line,x=X[i]),y=Y
[i])<=0)];
    end do;

    m:=0;
    for i from 2 to numelems(sign_list) while m=0 do
     TrueFalse:=evalb(sign_list[1]=sign_list[i]);
     if TrueFalse=false then
      m:=m+1;
     end if;
    end do;
    if m=0 then
     L_normdir:=[op(L_neg[1]),L_red[1]];
    elif m>0 then
     L_normdir:=[];
    end if;

   end if;

  elif numelems(L_neg)>1 then
   L_normdir:=[];

  end if;

 end if;

 return L_normdir;
```

```
  ### Output: List of extremals L_normdir that generate the normal
 cone

 end proc;
```

```
NormalCone := proc(A)                                                                        (4)
    local L_red, L_neg, L_normdir, X, Y, i, j, line, X0, Y0, sign1, sign2, sign_list, m, TrueFalse;
    L_red := OpposingPairs(A)[1];
    L_neg := OpposingPairs(A)[2];
    L_normdir := [ ];
    if numelems(L_red) = 1 then
        L_normdir := L_red
    else
        if numelems(L_neg) = 0 then
            X := [seq(L_red[i][1], i = 1 ..numelems(L_red))];
            Y := [seq(L_red[i][2], i = 1 ..numelems(L_red))];
            for i to numelems(X) while numelems(L_normdir) < 2 do
                line := x * Y[i] − y * X[i];
                X0 := subsop(i = NULL, X);
                Y0 := subsop(i = NULL, Y);
                sign1 := evalb(eval(eval(line, x = X0[1]), y = Y0[1]) <= 0);
                m := 0;
                for j from 2 to numelems(X0) while m = 0 do
                    sign2 := evalb(eval(eval(line, x = X0[j]), y = Y0[j]) <= 0);
                    if sign1 <> sign2 then m := m + 1 end if
                end do;
                if m = 0 then L_normdir := [op(L_normdir), [X[i], Y[i]]] end if
            end do
        elif numelems(L_neg) = 1 then
            if numelems(L_red) = 2 then
                L_normdir := L_red
            elif 3 <= numelems(L_red) then
                for i to numelems(L_neg[1]) do
                    L_red := remove(is, L_red, L_neg[1][i])
                end do;
                X := [seq(L_red[i][1], i = 1 ..numelems(L_red))];
                Y := [seq(L_red[i][2], i = 1 ..numelems(L_red))];
                line := x * L_neg[1][1][2] − y * L_neg[1][1][1];
                sign_list := [ ];
                for i to numelems(X) do
                    sign_list := [op(sign_list), evalb(eval(eval(line, x = X[i]), y = Y[i])
                     <= 0)]
                end do;
                m := 0;
```

```
              for i from 2 to numelems(sign_list) while m = 0 do
                  TrueFalse := evalb(sign_list[1] = sign_list[i]); if
                  TrueFalse = false then
                      m := m + 1
                  end if
              end do;
              if m = 0 then
                  L_normdir := [op(L_neg[1]), L_red[1]]
              elif 0 < m then
                  L_normdir := [ ]
              end if
          end if
      elif 1 < numelems(L_neg) then
          L_normdir := [ ]
      end if
  end if;
  return L_normdir
end proc
```

```
> Plotting:=proc(c,A,b,PlotIntervals)
   local L_normdir,FeasibleSet,FeasiblePlot,PointListPlot,
  ObjectiveArrow,NormDirPlot;

   ### Creating the figure for the problem
   ### Input:
   ## c: coefficients of the objective function in list form
   ## A: list of normal vectors generated by the rows of A
   ## b: list of the b_i constants from the right-hand sight of the
  constraints
   ## PlotIntervals: list of endpoints of the intervals for the
  plotting -> x=a..b,y=c..d -> PointIntervals:=[a,b,c,d]

   L_normdir:=NormalCone(A);

   FeasibleSet:=[seq(A[i][1]*x+A[i][2]*y<=b[i],i=1..numelems(A))];
   FeasiblePlot:=inequal(FeasibleSet,x=PlotIntervals[1]..
  PlotIntervals[2],y=PlotIntervals[3]..PlotIntervals[4],color=
  "LightGray"):
   PointListPlot:=pointplot([seq(A[i],i=1..numelems(A))],
  symbolsize=15,symbol=solidcircle,color="Blue");
   ObjectiveArrow:=arrow(c,width=0.001,head_width=0.2,head_length =
  0.2,color="Green",shape=arrow,scaling=constrained);

   if numelems(L_normdir)=0 then
    display(FeasiblePlot,ObjectiveArrow,PointListPlot,labels=
  ["x_1","x_2"]);
   else
    NormDirPlot:=arrow([seq(L_normdir[i],i=1..numelems(L_normdir))
  ],width=0.001,head_width=0.2,head_length = 0.2,color="Red",shape=
  arrow,scaling=constrained);
    printf("The normal cone is the cone combinatin of: %a.",
```

```
L_normdir);
  display(NormDirPlot,FeasiblePlot,ObjectiveArrow,PointListPlot,
labels=["x_1","x_2"]);
 end if;
end proc;
```

```
Plotting := proc(c, A, b, PlotIntervals)                                                    (5)
   local L_normdir, FeasibleSet, FeasiblePlot, PointListPlot, ObjectiveArrow, NormDirPlot;
   L_normdir := NormalCone(A);
   FeasibleSet := [seq(x*A[i][1] + y*A[i][2] <= b[i], i = 1 ..numelems(A))];
   FeasiblePlot := plots:-inequal(FeasibleSet, x = PlotIntervals[1] ..PlotIntervals[2], y
    = PlotIntervals[3] ..PlotIntervals[4], color = "LightGray");
   PointListPlot := plots:-pointplot([seq(A[i], i = 1 ..numelems(A))], symbolsize = 15, symbol
    = solidcircle, color = "Blue");
   ObjectiveArrow := plots:-arrow(c, width = 0.001, head_width = 0.2, head_length = 0.2, color
    = "Green", shape = plots:-arrow, scaling = constrained);
   if numelems(L_normdir) = 0 then
       plots:-display(FeasiblePlot, ObjectiveArrow, PointListPlot, labels = ["x_1", "x_2"])
   else
       NormDirPlot := plots:-arrow([seq(L_normdir[i], i = 1 ..numelems(L_normdir))],
       width = 0.001, head_width = 0.2, head_length = 0.2, color = "Red", shape = plots:-
       arrow, scaling = constrained);
       printf("The normal cone is the cone combinatin of: %a.", L_normdir);
       plots:-display(NormDirPlot, FeasiblePlot, ObjectiveArrow, PointListPlot, labels
        = ["x_1", "x_2"])
   end if
end proc
```

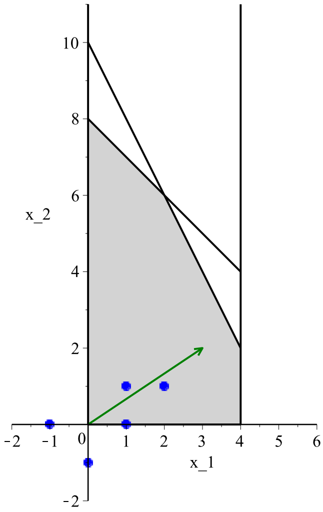
```
> Solvability:=proc(c,A,b,PlotIntervals)
   local L_normdir,la,line,sign_c,sign_3;

   ### Input: same as in the procedure Plotting

   L_normdir:=NormalCone(A);

   if numelems(L_normdir)=0 then
    printf("The normal cone is R^2. If the problem is feasible,
then the feasible set is compact. Thus, the problem has an
optimal solution, as well.");
    Plotting(c,A,b,PlotIntervals);

   elif numelems(L_normdir)=1 then
    la:=solve({la1*L_normdir[1][1]=c[1],la1*L_normdir[1][2]=c[2]},
{la1});
    if la=NULL then
     printf("The objective function is unbounded, therefore there
is no optimal solution.");
     Plotting(c,A,b,PlotIntervals);
    elif rhs(la[1])>=0 then
     printf("If the problem is feasible, then it has an optimal
solution, as well.");
```

```
   Plotting(c,A,b,PlotIntervals);
  else
   printf("The objective function is unbounded, therefore there
is no optimal solution.");
   Plotting(c,A,b,PlotIntervals);
  end if;

 elif numelems(L_normdir)=2 then
  if L_normdir[1][1]=0 and L_normdir[2][1]=0 and c[1]<>0 then
   printf("The objective function is unbounded, therefore there
is no optimal solution.");
   Plotting(c,A,b,PlotIntervals);
  elif L_normdir[1][2]=0 and L_normdir[2][2]=0 and c[2]<>0 then
   printf("The objective function is unbounded, therefore there
is no optimal solution.");
   Plotting(c,A,b,PlotIntervals);
  elif L_normdir[1][1]=0 and L_normdir[2][1]=0 and c[1]=0 or
L_normdir[1][2]=0 and L_normdir[2][2]=0 and c[2]=0 then
   printf("If the problem is feasible, then it has an optimal
solution, as well.");
   Plotting(c,A,b,PlotIntervals);
  else
   la:=solve({seq(la1*L_normdir[1][i]+la2*L_normdir[2][i]=c[i],i=
1..numelems(c))},{la1,la2});
   if rhs(la[1])<0 or rhs(la[2])<0 then
    printf("The objective function is unbounded, therefore there
is no optimal solution.");
    Plotting(c,A,b,PlotIntervals);
   else
    printf("If the problem is feasible, then it has an optimal
solution, as well.");
    Plotting(c,A,b,PlotIntervals);
   end if;
  end if;

 elif numelems(L_normdir)=3 then
  line:=L_normdir[1][2]*x-L_normdir[1][1]*y;
  sign_c:=evalb(eval(eval(line,x=c[1]),y=c[2])<=0);
  sign_3:=evalb(eval(eval(line,x=L_normdir[3][1]),y=L_normdir[3]
[2])<=0);
  if sign_c=sign_3 then
   printf("If the problem is feasible, then it has an optimal
solution, as well.");
   Plotting(c,A,b,PlotIntervals);
  else
   printf("The objective function is unbounded, therefore there
is no optimal solution.");
   Plotting(c,A,b,PlotIntervals);
  end if;

 end if;

### Output: Answer for the solvability question; the list
L_normdir; figure for the problem

end proc;
```

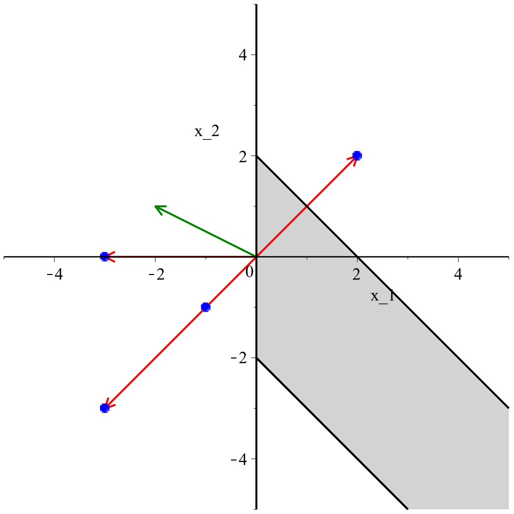
(6)

```
Solvability := proc(c, A, b, PlotIntervals)                                              (6)
    local L_normdir, la, line, sign_c, sign_3;
    L_normdir := NormalCone(A);
    if numelems(L_normdir) = 0 then
        printf("The normal cone is R^2. If the problem is feasible, then the feasible set is
        compact. Thus, the problem has an optimal solution, as well.");
        Plotting(c, A, b, PlotIntervals)
    elif numelems(L_normdir) = 1 then
        la := solve({la1 * L_normdir[1][1] = c[1], la1 * L_normdir[1][2] = c[2]}, {la1});
        if la = NULL then
            printf("The objective function is unbounded, therefore there is no optimal solution.
            ");
            Plotting(c, A, b, PlotIntervals)
        elif 0 <= rhs(la[1]) then
            printf("If the problem is feasible, then it has an optimal solution, as well.");
            Plotting(c, A, b, PlotIntervals)
        else
            printf("The objective function is unbounded, therefore there is no optimal solution.
            ");
            Plotting(c, A, b, PlotIntervals)
        end if
    elif numelems(L_normdir) = 2 then
        if L_normdir[1][1] = 0 and L_normdir[2][1] = 0 and c[1] <> 0 then
            printf("The objective function is unbounded, therefore there is no optimal solution.
            ");
            Plotting(c, A, b, PlotIntervals)
        elif L_normdir[1][2] = 0 and L_normdir[2][2] = 0 and c[2] <> 0 then
            printf("The objective function is unbounded, therefore there is no optimal solution.
            ");
            Plotting(c, A, b, PlotIntervals)
        elif L_normdir[1][1] = 0 and L_normdir[2][1] = 0 and c[1] = 0 or L_normdir[1]
        [2] = 0 and L_normdir[2][2] = 0 and c[2] = 0 then
            printf("If the problem is feasible, then it has an optimal solution, as well.");
            Plotting(c, A, b, PlotIntervals)
        else
            la := solve({seq(la1 * L_normdir[1][i] + la2 * L_normdir[2][i] = c[i], i = 1
             ..numelems(c))}, {la1, la2});
            if rhs(la[1]) < 0 or rhs(la[2]) < 0 then
                printf("The objective function is unbounded, therefore there is no optimal
                solution.");
                Plotting(c, A, b, PlotIntervals)
            else
```

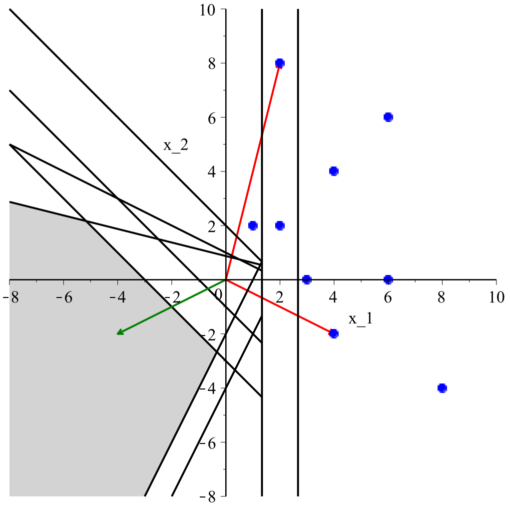
```
                printf("If the problem is feasible, then it has an optimal solution, as well.");
                Plotting(c, A, b, PlotIntervals)
            end if
        end if
    elif numelems(L_normdir) = 3 then
        line := x * L_normdir[1][2] − y * L_normdir[1][1];
        sign_c := evalb(eval(eval(line, x = c[1]), y = c[2]) <= 0);
        sign_3 := evalb(eval(eval(line, x = L_normdir[3][1]), y = L_normdir[3][2]) <= 0);
        if sign_c = sign_3 then
            printf("If the problem is feasible, then it has an optimal solution, as well.");
            Plotting(c, A, b, PlotIntervals)
        else
            printf("The objective function is unbounded, therefore there is no optimal solution.
            ");
            Plotting(c, A, b, PlotIntervals)
        end if
    end if
end proc
> ### Example 1 ###

  c:=[3,2]:
  A:=[[2,1],[1,1],[1,0],[-1,0],[0,-1]]:
  b:=[10,8,4,0,0]:
  PlotInterval:=[-2,6,-2,11]:
  Solvability(c,A,b,PlotInterval);
The normal cone is R^2. If the problem is feasible, then the
feasible set is compact. Thus, the problem has an optimal
solution, as well.
```



```
> ### Example 2 ###

  c:=[-2,1]:
  A:=[[-3,-3],[-3,0],[-1,-1],[2,2]]:
  b:=[6,0,2,4]:
  PlotInterval:=[-5,5,-5,5]:
  Solvability(c,A,b,PlotInterval);
If the problem is feasible, then it has an optimal solution, as
well.The normal cone is the cone combinatin of: [[-3, -3], [2,
2], [-3, 0]].
```



```
> ### Example 3 ###

  c:=[-4,-2]:
  A:=[[2,8],[2,2],[3,0],[4,-2],[4,4],[6,0],[8,-4],[1,2],[6,6]]:
  b:=[7,-6,4,8,-4,16,8,2,12]:
  PlotInterval:=[-8,10,-8,10]:
  Solvability(c,A,b,PlotInterval);
The objective function is unbounded, therefore there is no
optimal solution.The normal cone is the cone combinatin of: [[2,
8], [4, -2]].
```


10
8
6
x_2
4
2
-8
-6
-4
-2
0
2
4
6
8
10
x_1
-2
-4
-6
-8