Institute of Applied Mathematics
Middle East Technical University

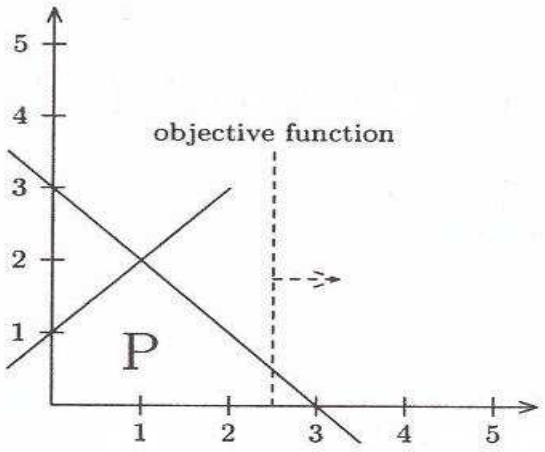
LECTURE NOTES SERIES

1

NUMERICAL OPTIMIZATION

Constrained Optimization

Bülent Karasözen
Gerhard-Wilhelm Weber



Design & Layout: Ömür Ugur

Preface

These are lecture notes offered to the students of the course *Numerical Optimization* at the *Institute of Applied Mathematics (IAM)* of *Middle East Technical University (METU)* in Summer Semester 2003. In these months, this course was held for the first time at our new and for Turkey pioneering institute which was founded in Autumn 2002. There has been the institute teacher's conviction that optimization theory is an important key technology in many modern fields of application from science, engineering, operational research and economy and, forthcoming, even in social sciences.

To be more precise, these lecture notes are prepared on the course's second part which treated the case of *constrained* continuous optimization from the numerical viewpoint. Here, we pay attention to both the cases of linear and nonlinear optimization (or: programming). In future, extensions of these notes are considered, especially in direction of unconstrained optimization. Herewith, our lecture notes are much more a service for the students than a complete book. They essentially are a *selection and a composition of three textbooks' elaborations*: There are the works "Lineare und Netzwerkoptimierung. Linear and Network Optimization. Ein bilinguales Lehrbuch" by *H. Hamacher and K. Klamroth* (2000) used in the parts about linear programming, "Linear and Nonlinear Programming" by *S.G. Nash and A. Sofer* (1996) and "Numerical Optimization" by *J. Nocedal and S.J. Wright* (1999) used for the parts about foundations and nonlinear programming.

During Summer Semester 2003, these lecture notes were given to the students in the handwritten form of a manuscript. We express our deep gratitude to *Dr. Ömür Uğur* from IAM of METU for having prepared this LaTeX-typed version with so much care and devotion. Indeed, we are looking forward that in future many of students of IAM will really enjoy these notes and benefit from them a lot. Moreover, we thank the scientific and institutional partners of IAM very much for the financial support which has made the lecture notes in the present form possible.

With friendly regards and best wishes,

Bülent Karasözen and Gerhard-Wilhelm Weber,
Ankara, in October 2003

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter with its various aspects of consideration, we take into account constraints additionally to our problem of minimizing an objective function f. Actually, we become concerned with the problem

$$\text{(P)} \qquad \begin{cases} \text{minimize} \quad f(x) \\ \text{subject to} \\ \quad h_i(x) = 0 \qquad \forall i \in I := \{1, 2, \ldots, m\}, \\ \quad g_j(x) \geq 0 \qquad \forall j \in J := \{1, 2, \ldots, s\}. \end{cases}$$

Here, f, h_i and g_j are supposed to be *smooth* real-valued function on $\mathbb{R}^n$. By smooth, we usually think of being one- or two-times continuously differentiable. The first group of constraints, where we demand $\ldots = 0$, are called *equality constraints*. The second group of constraints, where we ask $\ldots \geq 0$, are called *inequality constraints*. Denoting the feasible set, where we restrict the objective function f on, by

$$M := \left\{ x \ \in \mathbb{R}^n \ \middle| \ h_i(x) = 0 \ (i \in I), \ g_j(x) \geq 0 \ (j \in J) \right\},$$

our constrained optimization problem can be written as follows:

$$\text{(P)} \qquad \text{minimize} \quad f(x) \qquad \text{subject to} \quad x \in M$$

or equivalently,

$$\text{(P)} \qquad \min_{x \in M} f(x).$$

Depending on the context, namely on the various assumptions on f, h, g, I and J we make, we shall sometimes denote the functions, the problem and its feasible set a bit differently or specific. For example, this will be the case when f, h, g are linear (to be more precise: linearly affine). In fact, we are going to distinguish the linear and the nonlinear case, speak about linear optimization and nonlinear optimization. As in our course, the practical character and aspect of optimization is more emphasized than the analytical or topological ones, we also talk about (non-)linear programming. This may remind us of both the motivation of optimization by concrete applications, and the numerical solution algorithms. Because of the priority we gave to the numerical-algorithmical aspect, we make the following introductory Section not so long.

1.1 Some Preparations

By the following definition we extend corresponding notions from unconstrained optimization almost straightforwardly:

Definition 1.1. Let a vector (or, point) $x^* \in \mathbb{R}^n$ be given.

(i) We call x^* a *local solution* of (P), if $x^* \in M$ and there is a neighbourhood $\mathcal{N}(x^*)$ of x^* such that

$$f(x^*) \leq f(x) \qquad \forall x \in M \cap \mathcal{N}(x^*).$$

(ii) We call x^* a *strict local solution* of (P), $x^* \in M$ and there is a neighbourhood $\mathcal{N}(x^*)$ of x^* such that

$$f(x^*) < f(x) \qquad \forall x \in \left(M \cap \mathcal{N}(x^*)\right) \setminus \{x^*\}.$$

(iii) We call x^* an *isolated local solution* of (P), $x^* \in M$ and there is a neighbourhood $\mathcal{N}(x^*)$ of x^* such that x^* is the only local solution of (P) in $M \cap \mathcal{N}(x^*)$.

Sometimes, we also say (strict or isolated) *local minimizer* for a (strict or local) local solution.

Do you understand and geometrically distinguish these three conditions? We shall deepen our understanding in the exercises and in the following sections.

Example 1.1. Let us consider the problem

$$\text{(P)} \quad \begin{cases} \text{minimize} & \big(f(x) := \ \big) \ x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 - 2 = 0 \end{cases} \qquad \big(\text{i.e., } \ h(x) := x_1^2 + x_2^2 - 2\big),$$

Here, $|\, I \,| = \{1\}$ $(h_1 = h)$ and $J = \emptyset$. We see by inspection that the feasible set M is the circle of radius $\sqrt{2}$ centered at the origin (just the boundary of the corresponding disc, not its interior). The solution x^* is obviously $(-1, -1)^T$, this is the global solution of (P): From any other point on the circle, it is easy to find a way to move that stays feasible (i.e., remains on the circle) while decreasing f. For instance, from the point $(\sqrt{2}, 0)^T$ any move in the clockwise direction around the circle has the desired effect. Would you please illustrate this and the following graphically? Indeed, we see that at x^*, the gradient $\nabla h(x^*)$ is parallel to the gradient $\nabla f(x^*)$:

$$\Big(\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \Big) \quad \nabla f(x^*) = \lambda^* \nabla h(x^*)$$

where $\lambda^* = \lambda_1^* = -\frac{1}{2}$.

□

The condition found in the previous example to be necessary for a (local) minimizer (or, local minimum) can be expressed as follows:

$$\nabla_x L(x^*, \lambda^*) = 0,$$

where $L : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}$ is defined by $L(x, \lambda) := f(x) - \lambda h(x)$ and called the *Lagrange function*. The value λ^* is called the *Lagrange multiplier* (at x^*). This Lagrange multiplier rule is a first-order necessary optimality condition [1] (NOC) which can be extended to cases where I is of some other cardinality $m \leq n$. in fact, provided that the *Linear Independence Constraint Qualification* (a regularity condition) holds at x^*, saying that the gradients

$$\nabla h_1(x^*), \nabla h_2(x^*), \ldots, \nabla h_m(x^*),$$

regarded as a family (i.e., counted by multiplicity), are linearly independent, then there exist so-called Lagrange multipliers $\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*$ such that

$$\nabla f(x^*) = \sum_{i=1}^{m} \lambda_i^* \nabla h(x^*).$$

[1] In the next sections, we also consider feasibility to be a part of it.

The latter equation is just

$$\nabla_x L(x^*, \lambda^*) = 0,$$

where $L : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}$ is the so-called *Lagrange function* $L(x, \lambda) := f(x) - \lambda^T h(x)$, with $\lambda := (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ and $h := (h_1, h_2, \ldots, h_m)^T$.

Next, we slightly modify Example 1.1. We replace the equality constraint by an inequality.

Example 1.2. Let us consider the problem

$$\text{(P)} \quad \begin{cases} \text{minimize} & \big(f(x) := \ \big) \ x_1 + x_2 \\ \text{subject to} & 2 - x_1^2 - x_2^2 \geq 0 \end{cases} \qquad \big(\text{i.e., } \ g(x) := 2 - x_1^2 - x_2^2\big),$$

here, $I = \emptyset$ and $|\, J \,| = 1$ ($g = g_1$). The feasible set consists of the circle considered in Example 1.1 and its interior. We note that the gradient $\nabla g(x) = (-2x_1, -2x_2)^T$ points from the circle in direction of the interior. For example, inserting the point $x = (\sqrt{2}, 0)^T$ gives $\nabla g(x) = (-2\sqrt{2}, 0)^T$, pointing in the direction of the negative x_1- axis, i.e., inwardly (regarded from x). By inspection, we see that the solution of (P) is still $x^* = (-1, -1)^T$, and we have now

$$\nabla f(x^*) = \mu^* \nabla g(x),$$

where $\mu^* = \mu_1^* = \frac{1}{2} \geq 0$.

Again, we can write our necessary equation by using a Lagrange function, namely, $L(x, \mu) := f(x) - \mu g(x)$:

$$\nabla_x L(x^*, \mu^*) = 0, \quad \text{where} \quad \mu^* \geq 0.$$

Again, our necessary optimality condition can be generalized by admitting another finite cardinality s of J, and $m = |\, I \,|$ possibly to be positive ($m \leq n$). For this purpose, we need a regularity condition (constraint qualification) at our local minimizer x^*, once again. In the presence of both equality and inequality constraints, we can again formulate LICQ. For this purpose, we introduce the set of active inequality constraints at some feasible point x:

$$J_0(x) := \big\{ j \in J \big| g_j(x) = 0 \big\}.$$

Then, LICQ at x^* means the linear independence of the vectors (together regarded as a family)

$$\nabla h_i(x^*) \quad (i \in I), \qquad \nabla g_j(x^*) \quad \big(j \in J_0(x^*)\big)$$

(so that the gradients of the active inequalities are taken into account to the gradients of the equality constraints, additionally). Another famous regularity condition is somewhat weaker than LICQ and called MFCQ (Mangasarian-Fromovitz Constraint Qualification). The geometrical meaning of MFCQ is the following: At each point of M we have an inwardly pointing direction (relative to the surface given by the zero set of h). In the sequel, we refer to LICQ for simplicity. Now, our first-order necessary optimality conditions (NOC) are called *Karush-Kuhn-Tucker conditions*: There exist $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)^T$ and $\mu^* = \big(\mu_j^*\big)_{j \in J_0(x^*)}$ such that

$$\left\{ \begin{array}{rcll} \nabla f(x^*) & = & \displaystyle\sum_{i=1}^{m} \lambda_i^* \nabla h_i(x^*) + \sum_{j \in J_0(x^*)} \mu_j^* \nabla g_j(x^*), & \\ \mu_j^* & \geq & 0 \qquad \forall j \in J_0(x^*). & \end{array} \right.$$

The second ones of the Lagrange multipliers, namely μ_j^* $(j \in J_0(x^*))$, necessarily are nonnegative. Here, we may interpret this as by the existence of (relatively) inwardly pointing direction along which f does not decrease, when starting from our local minimizer x^*. Let us summarize our consideration by referring to the Lagrange function $L(x, \lambda, \mu) := f(x) - \lambda^T h(x) - \mu^T g(x)$, where $\big(\mu_j^*\big)_{j \in J_0(x^*)}$ is filled up by additional parameters μ_j $(j \notin J_0(x^*))$ (at $x = x^*$, we have $\mu_j^* = 0, j \notin J_0(x^*)$).

Theorem 1.1. *Suppose that x^* is a local solution of* (P)*, and that LICQ holds at x^*. Then, there uniquely exist Lagrange multiplier vectors $\lambda^* \in \mathbb{R}^m, \mu^* \in \mathbb{R}^s$, such that the following conditions are satisfied:*

(NOC)
$$\left\{ \begin{array}{rcl} \nabla_x L(x^*, \lambda^*, \mu^*) & = & 0, \\ h(x^*) & = & 0, \\ g(x^*) & \geq & 0, \\ \mu^* & \geq & 0, \\ \mu_j^* g_j(x^*) & = & 0 \ \forall j \in J. \end{array} \right.$$

The latter multiplicative conditions are local complementarity conditions. To which case study do they give rise?

Proof. See Nocedal, Wright (1999); the proof of uniqueness is left to the reader. □

Often, the compact form of (NOC), where we do not explicitly refer to the active inequalities $j \in J_0(x^*)$, is more convenient. In other times, a direct reference to $J_0(x^*)$ is better for us. We note that our first-order necessary optimality conditions are not the only ones. In the subsequent sections and in the presentation given in lectures, we shall briefly indicate them sometimes. These conditions are, more or less, generalization of the condition (taught to us in the unconstrained case) that the gradient vanishes at a local solution: $\nabla f(x^*) = 0$, if $I = J = \emptyset$. Provided that all the defining functions are twice continuously differentiable, then we know the second-order optimality condition saying that the Hessian matrix has to be positive semi-definite:

$$\rho^T \nabla^2_{xx} f(x^*) \rho \geq 0 \qquad \forall \rho \in \mathbb{R}^n.$$

But how does this condition look like in our case of constrained optimization? Let us define the tangent space of M at x^* as follows:

$$T_{x^*} := \left\{ \rho \in \mathbb{R}^n \middle| \nabla^T h_i(x^*)\rho = 0 \ \forall i \in I, \quad \nabla^T g_j(x^*)\rho = 0 \ \forall j \in J_0(x^*) \right\}.$$

Then, our second-order condition states that the Hessian of the Lagrange function L at x^* is positive semi-definite over T_{x^*}. Here, we take the vectors $\rho \in \mathbb{R}^n$ for left- or right-multiplication from T_{x^*}. This can also be expressed as the positive semi-definiteness of this Hessian, when firstly left- and right-multiplied by B^T and B, respectively, over $\mathbb{R}^n$. Here, B is any matrix whose columns constitute a basis of the linear space T_{x^*}. Herewith, we can express our second-order optimality condition so, where $\widehat{n} = n - \widehat{m}$, $\widehat{m} = |\, J_0(x^*)\,|$:

$$(\text{NOC})_{s.o.} \qquad \tilde{\rho}^T B^T \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) B \tilde{\rho} \geq 0 \quad \forall \tilde{\rho} \in \mathbb{R}^{\widehat{n}}.$$

In literature, we also find some refined condition, where in the statement of positive-definiteness the reference space T_{x^*} is replaced by the set $C^\bullet_{x^*}$ which is a (tangent) cone. Namely, $C^\bullet_{x^*}$ comes from substituting the gradient conditions on the active inequalities by

$$\nabla^T g_j(x^*)\rho = 0 \qquad \forall j \in J_0(x^*) \text{ with } \mu^*_j > 0,$$

$$\nabla^T g_j(x^*)\rho \geq 0 \qquad \forall j \in J_0(x^*) \text{ with } \mu^*_j = 0.$$

Theorem 1.2. *Suppose that x^* is a local solution of* (P)*, and that LICQ is satisfied at x^*. Let λ^* and μ^* be the corresponding Lagrange multiplier vectors so that* (NOC) *is (necessarily) fulfilled.*

Then, the Hessian $\nabla^2_{xx} L(x^*, \lambda^*, \mu^*)$ is positive semi-definite over T_{x^*}, or (refined version) over $C^\bullet_{x^*}$.

Proof. See, e.g., Nash, Sofer (1996) and Nocedal, Wright (1999). □

This theorem plays an important role for detecting that some found or submitted stationary point x^* (i.e., x^* fulfills our first-order necessary optimality conditions) is not a local solution, but a local minimizer or a saddle point. How can we use Theorem 1.2 for this purpose? A saddle point is in between of a local minimizer and a local maximizer: there are feasible directions of both an increasing and a decreasing behaviour of f. But how can we detect that a stationary point, i.e., a candidate for being a local minimizer really is such a one? what we know from the unconstrained case gives rise to assume that we should turn from positive semi-definiteness (over T_{x^*} or $C^\bullet_{x^*}$) to positive definiteness (over $T_{x^*} \setminus \{0\}$ or $C^\bullet_{x^*} \setminus \{0\}$). In fact, by this sharpening of $(\mathrm{NOC})_{s.o.}$ we obtain a condition which together with our first-order (NOC) is sufficient for x^* to become a local solution. In particular, we arrive at

$$(\mathrm{SOC})_{s.o.} \qquad \tilde{\rho}^T B^T \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) B\tilde{\rho} > 0 \quad \forall \tilde{\rho} \in \mathbb{R}^{\widehat{n}} \setminus \{0\}\,.$$

We state the following theorem on second-order sufficient optimality conditions:

Theorem 1.3. *Suppose that for some point $x^* \in M$ there are Lagrange multiplier vectors λ^*, μ^* such that the conditions* (NOC) *(of first-order) are satisfied and $\mu^*_j > 0$ $(j \in J_0(x^*))$. Suppose also that the Hessian $\nabla^2_{xx} L(x^*, \lambda^*, \mu^*)$ is positive definite over T_{x^*}, or over $C^\bullet_{x^*}$ (i.e., over $T_{x^*} \setminus \{0\}$, or $C^\bullet_{x^*} \setminus \{0\}$).*

Then, x^* is a strict local solution for (P)*.*

Proof. See, e.g., Nash, Sofer (1996) and Nocedal, Wright (1999). □

In our exercises, we shall consider a few examples on finding local or global solutions for nonlinear optimization problem, where the utilization of numerical-algorithmical methods is still not necessary. Let us also conclude our Section 1.1 with another such example.

Example 1.3. Let us consider the problem (P) from Example 1.2:

$$\text{(P)} \quad \begin{cases} \text{minimize} & \big(f(x) := \ \big) \ x_1 + x_2 \\ \text{subject to} & 2 - x_1^2 - x_2^2 \geq 0 \end{cases} \qquad \big(\text{i.e., } \ g(x) := 2 - x_1^2 - x_2^2\big).$$

We want to check the second-order conditions. The Lagrangian is

$$L(x, \mu) := (x_1 + x_2) - \mu(2 - x_1 - x_2),$$

and it is easy to show that the Karush-Kuhn-Tucker conditions (NOC) are satisfied by $x^* = (-1, -1)^T$, with $\mu^* = \frac{1}{2}$. The Hessian of L (with respect to x) is

$$\nabla^2_{xx} L(x^*, \mu^*) = \begin{pmatrix} 2\mu^* & 0 \\ 0 & 2\mu^* \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

This matrix is positive definite, i.e., it satisfies $\rho^T \nabla^2_{xx} L(x^*, \mu^*)\rho > 0$ for all $\rho \in \mathbb{R}^n \setminus \{0\}$ (in particular, for all elements of $T_{x^*}, C^{\bullet}_{x^*}$, except 0). So, it certainly satisfies the conditions of Theorem 1.3.

We conclude that $x^* = (-1, -1)^T$ is a strict local solution for (P). Let us remark that x^* is a even a global solution, since (P) is a convex programming problem. □

In this course, we do not pay much attention to stability aspects which, however, can be well analyzed by LICQ and $(\text{SOC})_{s.o}$. Next, we concentrate of the linear case of (P).

Chapter 2

Linear Programming: Foundations and Simplex Method

In this chapter, based on the book of Hamacher, Klamroth (2000), we consider easy examples which lead to optimization ("programming") problems: Linear programs play a central role in modelling of optimization problems.

Example 2.1. The manufacturer "Chocolate & Co." is in the process of reorganizing its production. Several chocolate products which have been produced up to now are taken out off production and are replaced by two new products. The first product P1 is fine cocoa, and the second one P2 is dark chocolate. The company uses three production facilities F1, F2, and F3 for the production of the two products. In the central facility F1 the cocoa beans are cleaned, roasted and cracked. In F2, the fine cocoa and in F3 the dark chocolate are produced from the preprocessed cocoa beans. Since the chocolate products of the company are known for their high quality, it can be assumed that the complete output of the company can be sold on the market. The profit per sold production unit of P1 (50kg of cocoa) is 3 (€), whereas it is 5 per sold production unit of P2 (100kg of chocolate). However, the capacities of F1–F3 are limited as specified in Table 2.1.

(For example, the row corresponding to F1 implies that the per unit P1 and P2, 3% and 2%, respectively, of the daily capacity of production facility F1 are needed, and that 18% of the daily capacity of F1 is available for the production of P1 and P2.)

The problem to be solved is to find out how many units x_1 of products P1 and x_2 of product P2 should be produced per day in order to maximize

	P1	P2	available capacity (in % of the daily capacity)
F1	3	2	18
F2	1	0	4
F3	0	2	12

Table 2.1: Chocolate production.

the profit (while satisfying the capacity constraints.) This problem can be formulated as a linear program (LP):

$$\text{(LP)} \qquad \begin{cases} \text{maximize} \quad 3x_1 + 5x_2 =: c^T x \\ \text{subject to the constraints} \\ \qquad 3x_1 + 2x_2 \leq 18 \qquad \text{(I)} \\ \qquad x_1 \qquad\quad \leq 4 \qquad \text{(II)} \\ \qquad\quad 2x_2 \leq 12 \qquad \text{(III)} \\ \qquad x_1, \quad x_2 \geq 0 \qquad \text{(IV), (V)} \end{cases}$$

The function $c^T x$ is the (linear) objective function of the LP. The constraints are partitioned into functional constraints ((I)–(III)) and nonnegativity constraints ((IV),(V)). Each x which satisfies all the constraints, is called a feasible solution of the LP and $c^T x$ is its objective (function) value. Instead of maximizing the function, it is often minimized. In this case, the coefficients c_i of the objective function can be interpreted as unit costs. The functional constraints can also be equations ("=") or inequalities with "$\geq$".

Let us continue by introducing a graphical procedure for the solution of (LP). In a first step, we draw the set of feasible solutions (the feasible region P) of (LP), that is the set of all points (vectors) (x_1, x_2), in matrix notation: $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, satisfying all the constraints; see Figure 2.1.

If $A^i x \leq b_i$ is one of the constraints (e.g., $A^i = (3, 2),\ b_1 = 18$) we draw the line corresponding to the equation

$$A^i x = b_i.$$

This space separates the space $\mathbb{R}^2$ into two half-spaces

$$A^i x \leq b_i, \text{ and } A^i x \geq b_i$$

The set P of feasible solutions is a subset of the half-space $A^i x \leq b_i$. We obtain P by taking the intersection of all these half-spaces including the half-

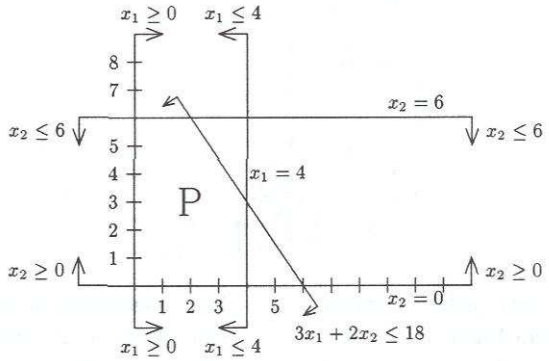


Figure 2.1: Simplex algorithm, feasible region.

spaces $x_1 \geq 0$ and $x_2 \geq 0$. A set of points in $\mathbb{R}^2$, which is obtained in such a way is called a *convex polyhedron*.

In a second step, we draw the objective function $z := c^T x = 3x_1 + 5x_2$. For any given value of z we obtain a line, and for any two different values of z we obtain two parallel lines for which z is as large as possible. The following Figure 2.2 shows the feasible region P and, additionally, the line corresponding to the objective function $3x_1 + 5x_2$, i.e., the line corresponding to the objective value $z = 15$. The arrow perpendicular to this line indicates, in which direction the line should be shifted in order to increase z.

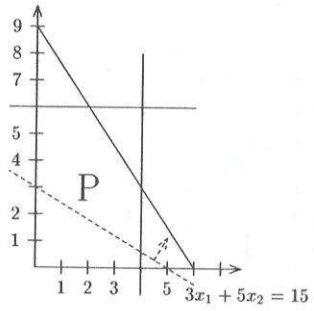


Figure 2.2: Simplex algorithm.

The intersection of any line $c^T x = z$ with P in some $x \in \mathrm{P}$ corresponds to a feasible solution $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ with objective value z. Hence, in order to maximize the objective function, the line is shifted parallel as far as possible without violating the condition

$$\left\{x \in \mathbb{R}^2 \middle| c^T x = z\right\} \cap \mathrm{P} \neq \emptyset.$$

This just gives the line which passes through the point $x^* = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$. Hence, $z^* = c^T x^* = 3 \cdot 2 + 5 \cdot 6 = 36$ is the maximum possible profit.

As a consequence, the company "Chocolate & Co." will produce 2 production units of cocoa and 6 production units of the dark chocolate per day. □

In this example, we observe an important property of linear programs: There is always an optimal solution (i.e., a feasible solution for which $c^T x$ is maximal) which is a corner point of P –or there is no optimal solution at all. This general property of LPs is a fundamental for the method introduced below as a general solution for LPs: the *simplex method*:

Imagine to move from corner point to corner point of P (a corner point is an element of P which is an intersection of two or more of its facets) using a procedure which guarantees that the objective value is improved in every step. As soon as we have reached a corner point in which no further improvement of the objective value is possible, the algorithm stops and the corner point corresponds to an optimal solution of the LP. In Example 2.1, such a sequence of corner points is, e.g., as stated in Table 2.2.

	corner point	objective value $3x_1 + 5x_2$
1.	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	0
2.	$\begin{pmatrix} 4 \\ 0 \end{pmatrix}$	12
3.	$\begin{pmatrix} 4 \\ 3 \end{pmatrix}$	27
4.	$\begin{pmatrix} 2 \\ 6 \end{pmatrix}$	36 (STOP), no further improvement possible.

Table 2.2: Corner points and values.

This is exactly the idea on which the simplex method is based, which was developed by G. Dantzig in 1947. It yields an optimal solution (whenever it exists) for only LP. An important step in the development of the preceding idea for arbitrary LPs is the application of methods from linear algebra.

Definition 2.1. A given LP is said to be in *standard form* if and only if it is given as

$$(\mathrm{LP})_{st} \qquad \left\{ \begin{array}{lrcl} \text{minimize} & c^T & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \quad (\text{i.e., } x_i \geq 0 \ \forall i \in \{1, \ldots, n\}), \end{array} \right.$$

where A in an $m \times n$ matrix, $m \leq n$, and $\mathrm{rank}(A) := \dim(\mathrm{Im}\, T) = m$ (T being the linear transformation represented by A relative to standard bases).

If $m \leq n$ or $\mathrm{rank}(A) = m$ are not the case, then we can reintroduce $\tilde{m} \leq n, \mathrm{rank}(A) =: \tilde{m}$ by omitting redundant (i.e., linearly on other $\tilde{m}$ ones depending) constraints. In the following, we show how any given LP can be transformed into standard form.

Assume that an LP in general form is given:

$$(\mathrm{LP})_{gf} \qquad \left\{ \begin{array}{lrcl} \text{minimize} & c_1 x_1 + \cdots + c_n x_n & & \\ \text{subject to} & a_{i1} x_1 + \cdots + a_{in} x_n & = & b_i \ (i \in \{1, \ldots, p\}) \\ & a_{i1} x_1 + \cdots + a_{in} x_n & \leq & b_i \ (i \in \{p+1, \ldots, q\}) \\ & a_{i1} x_1 + \cdots + a_{in} x_n & \geq & b_i \ (i \in \{q+1, \ldots, m\}) \\ & x_j & \geq & 0 \ (j \in \{1, \ldots, r\}) \\ & x_j & \leq & 0 \ (j \in \{r+1, \ldots, n\}) \end{array} \right.$$

(a) The "$\leq$" constraints can be transformed into equality constraints by introducing *slack variables*

$$x_{n+i-p} := b_i - a_{i1} x_1 - \cdots - a_{in} x_n \ (i \in \{p+1, \ldots, q\}).$$

We obtain

$$\left. \begin{array}{rcl} a_{i1} x_1 + \cdots + a_{in} x_n + x_{n+i-p} & = & b_i \\ x_{n+i-p} & \geq & 0 \end{array} \right\} \quad (i \in \{p+1, \ldots, q\}).$$

(b) Analogously, the "$\geq$" constraints can be transformed into equality constraints by introducing *surplus variables*

$$x_{n+i-p} := a_{i1} x_1 + \cdots + a_{in} x_n - b_i \ (i \in \{q+1, \ldots, m\}).$$

We obtain

$$\left. \begin{array}{rcl} a_{i1} x_1 + \cdots + a_{in} x_n - x_{n+i-p} & = & b_i \\ x_{n+i-p} & \geq & 0 \end{array} \right\} \quad (i \in \{q+1, \ldots, m\}).$$

(c) An LP in which the objective function is to be maximized can be transformed into standard form by using the identity

$$\max\left\{c^T x\middle|\ \cdots\right\} = -\min\left\{(-c)^T x\middle|\ \cdots\right\},$$

where "$\cdots$" stand for properties required for x, and by solving an LP with the coefficients $-c_j$ in standard form.

(d) If a variable x_j is not sign constrained (denoted by $x_j \gtrless 0$), then we replace it by

$$x_j =: x_j^+ - x_j^- \quad \text{with} \quad x_j^+, x_j^- \geq 0,$$

in order to transform the problem into standard form.

in connection with LPs, we will usually denote as follows (see $(\mathrm{LP})_{st}$):

$$A = \left(a_{ij}\right)_{\substack{i\in\{1,\ldots,m\}\\ j\in\{1,\ldots,n\}}} = \left(a_{ij}\right)\ :\ m \leq n,\ \mathrm{rank}(A) = m,$$

$$A_j = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix}, \qquad A^i = (a_{i1} \ldots a_{in}),$$

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix},$$

$$P = \left\{x \in \mathbb{R}^n \middle|\ Ax = b,\ x \geq 0\right\}.$$

Basic Solutions: Optimality Test and Basic Exchange

Definition 2.2. A basis of A is a set $\mathcal{B} := \left\{A_{B(1)}, \ldots, A_{B(m)}\right\}$ of m linear independent columns of A. The index set is given in an arbitrary but fixed order: $B := \big(B(1), \ldots, B(m)\big)$.

Often, we call (somewhat inexact) the index set B itself a basis of A.

By $A_B := \big(A_{B(1)}, \ldots, A_{B(m)}\big)$ we denote the regular $m \times m$ sub-matrix of A corresponding to B. The corresponding variables x_j are called *basic variables*, and they are collected in the vector

$$x_B := \begin{pmatrix} x_{B(1)} \\ \vdots \\ x_{B(m)} \end{pmatrix}.$$

The remaining indices are contained in a set which is again in arbitrary but fixed order: $N := \big(N(1), \ldots, N(n-m)\big)$. The variables x_j with $j \in N$ are called *non-basic variables*.

We immediately obtain

$$\left.\begin{array}{l} x \text{ is a solution of } Ax = b \Leftrightarrow \\ (x_B, x_N) \text{ is a solution of } A_B x_B + A_N x_N = b \\ \text{(i.e., } \begin{pmatrix} x_B \\ x_N \end{pmatrix} \text{ is a solution of } (A_B, A_N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b\text{).} \end{array}\right\}$$

(This can easily be seen by changing the order of the columns A_j suitably.) By multiplying the latter equation by A_B^{-1} we obtain:

$$\begin{array}{l} x \text{ is a solution of } Ax = b \Leftrightarrow \\ (x_B, x_N) \text{ is a solution of } x_B = A_B^{-1} x_B - A_B^{-1} A_N x_N; \quad (*) \end{array}$$

$(*)$ is called the *basic representation* of x (with respect to B).

Definition 2.3. For any choice of the non-basic variables $x_{N(j)}$ $(j \in \{1, \ldots, n-m\})$ we obtain, according to $(*)$, uniquely defined values of the basic variable $x_{B(j)}$ $(j \in \{1, \ldots, m\})$. The *basic solution* (with respect to B) is the solution of $Ax = b$ with $x_N = 0$ and, $x_B = A_B^{-1} b$.

A basic solution is called a *basic feasible solution* if and only if $x_B \geq 0$.

Example 2.2. We consider the LP

$$\text{(LP)} \qquad \left\{ \begin{array}{lrcl} \text{maximize} & x_1 & & \\ \text{subject to} & -x_1 + x_2 & \leq & 1 \\ & x_1 + x_2 & \leq & 3 \\ & x_1, x_2 & \geq & 0. \end{array} \right.$$

The feasible region is illustrated in Figure 2.3.

We can easily see that $x^* = (3, 0)^T$ is the optimal solution. First, we transform this LP into standard form by introducing slack variables x_3, x_4 and by transforming the maximization into a minimization (the "$-$" sign

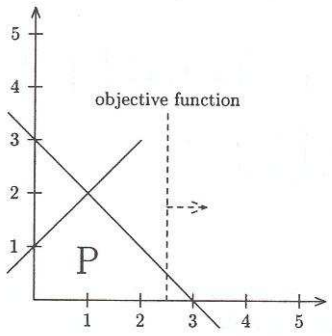


Figure 2.3: Feasible region.

in front of "min" can be dropped, but we must not forget this "−" when interpreting the final result). Hence we obtain

$$(\mathrm{LP})_{st} \qquad \begin{cases} \text{minimize} & -x_1 & & \\ \text{subject to} & -x_1 + x_2 + x_3 & = & 1 \\ & x_1 + x_2 + x_4 & = & 3 \\ & x_1, x_2, x_3, x_4 & \geq & 0, \end{cases}$$

i.e., $c = (-1, 0, 0, 0)^T,\ b = (1, 3)^T$ and

$$A = \begin{pmatrix} -1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

(i) For $B = (B(1), B(2)) = (3, 4)$ we obtain the basic solution

$$x_B = \begin{pmatrix} x_{B(1)} \\ x_{B(2)} \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = A_B^{-1} b = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} b = b = \begin{pmatrix} 1 \\ 3 \end{pmatrix},$$

$$x_N = \begin{pmatrix} x_{N(1)} \\ x_{N(2)} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0,$$

hence, $x = (0, 0, 1, 3)^T$.

(ii) For $B = (1, 2)$ we get by a proposition from linear algebra

$$A_B = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \Longrightarrow A_B^{-1} = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix},$$

therefore,

$$x_B = \begin{pmatrix} x_{B(1)} \\ x_{B(2)} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A_B^{-1} b = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

$$x_N = \begin{pmatrix} x_{N(1)} \\ x_{N(2)} \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = 0,$$

hence, $x = (1, 2, 0, 0)^T$.

(iii) For $B = (4, 1)$ we get (exercise)

$$x_B = \begin{pmatrix} x_4 \\ x_1 \end{pmatrix} = A_B^{-1} b = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ -1 \end{pmatrix},$$

$$x_N = \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = 0,$$

hence, $x = (-1, 0, 0, 4)^T$.

Evaluation: In the cases (i), (ii), the basic solutions correspond to corner points of P, namely, (i): x corresponds to the corner point $(0, 0)$, and (ii): x corresponds to the corner point $(1, 2)$. Since $x_B \geq 0$ in both cases, (i) and (ii) yield basic feasible solutions.

In the case (iii) however, $x_B \geq 0$ is not satisfied. So, (x_B, x_N) is a basic solution which is not feasible. In Figure 2.3, it can be easily seen that the corresponding point $(x_1, x_2) = (-1, 0)$ is not contained in P.

□

Now, we consider a basic feasible solution (x_B, x_N) and use the basic representation with respect to B to derive an optimality criterion. First, we partition the coefficients of c into

$$c_B := (c_{B(1)}, \ldots, c_{B(m)})^T \qquad \text{and}$$

$$c_N := (c_{N(1)}, \ldots, c_{N(n-m)})^T.$$

Then, the objective value $c^T x$ can be written as

$$\begin{aligned} c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T \left(A_B^{-1} b - A_B^{-1} A_N x_N\right) + c_N^T x_N \\ &= c_B^T A_B^{-1} b + \left(c_N^T - c_B^T A_B^{-1} A_N\right) x_N. \end{aligned}$$

In the current basic feasible solution we have that $x_N = 0$ and, therefore, its objective value is

$$c_B^T x_B = c_B^T A_B^{-1} b.$$

For all other solutions the objective value differs from this value by $\left(c_N^T - c_B^T A_B^{-1} A_N\right) x_N$. If we increase the value of $x_{N(j)} = 0$ to $x_{N(j)} = \delta,\ \delta > 0$, then the objective value is changed by

$$\delta \left(c_{N(j)} - \overbrace{\underbrace{c_B^T A_B^{-1}}_{=:\Pi} A_{N(j)}}^{=:z_{N(j)}}\right).$$

Therefore, the objective value of the given basic feasible solution increases if $c_{N(j)} - z_{N(j)} > 0$, and it decreases if $c_{N(j)} - z_{N(j)} < 0$. The values

$$\overline{c}_{N(j)} := c_{N(j)} - z_{N(j)},$$

called the *reduced* or *relative costs* of the non-basic variable $x_{N(j)}$, thus contain the information whether it is useful to increase the value of the non-basic variable $x_{N(j)}$ from 0 to a value $\delta > 0$. In particular, we obtain

Theorem 2.1 (Optimality Condition for Basic Feasible Solutions). *If x is a basic feasible solution with respect to B and if*

$$\overline{c}_{N(j)} = c_{N(j)} - z_{N(j)} = c_{N(j)} - c_B^T A_B^{-1} A_{N(j)} \geq 0 \ \forall j \in \{1, \ldots, n-m\},$$

then x is an optimal solution of the LP

$$(\mathrm{LP})_{st} \qquad \begin{cases} \text{minimize} & c^T & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0. \end{cases}$$

As we have seen in Example 2.2, basic feasible solutions of $Ax = b$ correspond to extreme points of the feasible region $\widehat{A}\widehat{x} \leq b$. Here, $\widehat{A}$ is derived from A by deleting those columns of A which correspond to a unit matrix,

and $\widehat{x}$ are the components of x which correspond to the columns of $\widehat{A}$. According to the idea mentioned at the beginning of this chapter, we will move to a new basic feasible solution if the optimality condition is not satisfied for the current basic feasible solution.

Example 2.3 *(Continuation of Example 2.2).* We apply our optimality test to two basic feasible solutions of Example 2.2:

(a) $B = (1, 2)$: In (ii) we computed that

$$A_B^{-1} = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Thus, we obtain:

$$\begin{array}{rcll} z_{N(1)} & = & \big(c_{B(1)}, c_{B(2)}\big) A_B^{-1} A_{N(1)} & \\ & = & (c_1, c_2) A_B^{-1} A_3 & \\ & = & (-1, 0) \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \\ & = & \frac{1}{2}(1, -1) \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \\ & = & \frac{1}{2} \qquad (\text{i.e., } \Pi = \frac{1}{2}(1, -1)), & \qquad N(1) = 3 \\ \Longrightarrow \ \overline{c}_{N(1)} & = & c_3 - z_3 = 0 - \frac{1}{2} < 0. & \end{array}$$

So, the optimality condition is violated. Note that, using Theorem 2.1 one cannot conclude that the corresponding basic feasible solution is not optimal, since our theorem gives only a sufficient optimality condition.

(b) $B = (1, 3)$:

$$A_B = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \implies A_B^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

We check the optimality condition by computing $\overline{c} := c_N - z_N$ with $z_N^T := c_B^T A_B^{-1} A_N$, and check whether $\overline{c}_N \geq 0$.

$$\begin{aligned} z_N^T &= c_B^T A_B^{-1} A_N \\ &= (c_1, c_3) A_B^{-1} (A_1, A_4) \\ &= (-1, 0) \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \\ &= (-1, 0) \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \\ &= (-1, -1) \\ \Longrightarrow \overline{c}_N &= (0,0)^T - (-1,-1)^T = (1,1)^T \geq 0. \end{aligned}$$

Hence, the basic feasible solution

$$x_B = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = A_B^{-1} b = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix},$$

$$x_N = \begin{pmatrix} x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

corresponding to B is optimal. This basic feasible solution corresponds to the corner point

$$x^* = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix},$$

which we have already identified in Example 2.2 as optimal using the graphical-geometrical procedure.

□

In the following, we show how to obtain a new feasible solution if the current one does not satisfy the optimality condition.

Suppose that $\overline{c}_{N(s)} = c_{N(s)} - z_{N(s)} < 0$ for some $s \in N$. Since

$$c^T x = c_B^T A_B^{-1} b + (c_N^T - C_B^T A_B^{-1} A_N) x_N$$

increasing $x_{N(s)}$ by one unit will decrease $c^T x$ by $\overline{c}_{N(s)}$. Since our goal is to minimize the objective function $c^T x$, we want to increase $x_{N(s)}$ by as many units as possible.

How large can $x_{N(s)}$ be chosen? This question is answered by the basic representation $(*)$ and the requirement $x_B \geq 0$. If we keep $x_{N(j)}$, $j \neq s$, equal to 0 and increase $x_{N(s)}$ from 0 to $\delta \geq 0$, then we obtain for the resulting solution of $Ax = b$

$$x_B = A_B^{-1}b - A_B^{-1}A_{N(s)}\delta.$$

Denoting the ith component of $A_B^{-1}b$ and $A_B^{-1}A_{N(s)}$ by $\tilde{b}_i$ and $\tilde{a}_{iN(s)}$, respectively, we have to choose δ such that

$$x_{B(i)} = \tilde{b}_i - \tilde{a}_{iN(s)}\delta \geq 0. \qquad (\otimes)$$

In order to choose δ as large as possible, we thus compute δ as

$$\delta = x_{N(s)} := \min\left\{ \frac{\tilde{b}_i}{\tilde{a}_{iN(s)}} \;\middle|\; \tilde{a}_{iN(s)} > 0 \right\}. \qquad \text{(min ratio rule)}$$

While computing δ with respect to the min ratio rule, two cases may occur:

Case 1: $\tilde{a}_{iN(s)} \leq 0 \; \forall i \in \{1, \ldots, m\}$.
Then, δ can be chosen arbitrarily large without violating any of the nonnegativity of constraints. As a consequence, $c^T x$ can be made arbitrarily small and the LP is unbounded. Thus we obtain:

Theorem 2.2 (Criterion on Unbounded LPs). *If x is a basic feasible solution with respect to B and if*

$$\overline{c}_{N(s)} < 0 \quad \textit{and} \quad A_B^{-1}A_{N(s)} \leq 0$$

for some $s \in N$, then the LP

$$(\mathrm{LP})_{st} \qquad \begin{cases} \text{minimize} & c^T & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{cases}$$

is unbounded.

Case 2: $\exists i \in \{1, \ldots, m\} : \tilde{a}_{iN(s)} > 0$.
In this case, the minimum in the computation of $x_{N(s)}$ by the min ratio rule is attained. Suppose that we get $\delta = x_{N(s)} = \dfrac{\tilde{b}_r}{\tilde{a}_{rN(s)}}$. (If the index r is not uniquely determined, then we choose any of the indices such that $\dfrac{\tilde{b}_r}{\tilde{a}_{rN(s)}} = \delta$.) According to $(\otimes)$ and the min ratio rule, the new solution is

$$x_{N(s)} = \frac{\tilde{b}_r}{\tilde{a}_{rN(s)}}, \qquad x_{N(j)} = 0 \quad \forall j \neq s,$$

$$x_{B(i)} = \tilde{b}_i - \tilde{a}_{iN(s)} x_{N(s)} = \tilde{b}_i - \tilde{a}_{iN(s)} \frac{\tilde{b}_r}{\tilde{a}_{rN(s)}} \qquad \forall i.$$

In particular, $x_{B(r)} = 0$. It is easy to check that

$$B' = \big(B'(1), \ldots, B'(r-1), B'(r), B'(r+1), \ldots, B'(m)\big)$$

with

$$B'(i) := \begin{cases} B(i) & \text{if } i \neq r \\ N(s) & \text{if } i = r \end{cases}$$

defines a new basis for A (argument: $\tilde{a}_{rN(s)} \neq 0$). Thus, the computation of $x_{N(s)}$ has induced a basis change. The variable $x_{B(r)}$ has left the basis, and $x_{N(s)}$ has entered the basis. The new index set of non-basic variables is

$$N' = \big(N'(1), \ldots, N'(s-1), N'(s), N'(s+1), \ldots, N'(n-m)\big)$$

with

$$N'(j) := \begin{cases} N(j) & \text{if } j \neq s \\ B(r) & \text{if } j = s. \end{cases}$$

Our basis change is indicated by Figure 2.4.

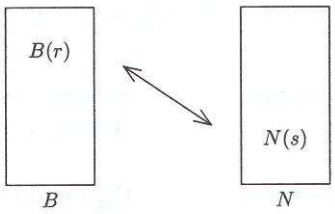


Figure 2.4: Basis change.

Example 2.4 *(Continuation of Example 2.3).* In (a), we saw that the optimality condition is violated for $B = (1, 2)$. Since $\overline{c}_3 < 0$, we want to let $x_{N(1)} = x_3$ enter into the basis. We compute (cf. Example 2.3):

$$\begin{pmatrix} \tilde{a}_{1N(1)} \\ \tilde{a}_{2N(1)} \end{pmatrix} = A_B^{-1} A_{N(1)} = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1 \\ 1 \end{pmatrix},$$

$$\tilde{b} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix} = A_B^{-1} b = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Longrightarrow \quad x_{N(1)} = x_3 = \frac{\tilde{b}_2}{\tilde{a}_{2N(1)}} = 4.$$

(Note that here the minimization of the min ratio rule is taken over a single-element set.) Thus, we obtain $B' = (1, 3)$ as our new index set of the basis (x_2 has left the basis, x_3 has entered the basis). The corresponding basic feasible solution is

$$\begin{aligned} x_{B'(2)} &= x_{N(1)} = x_3 = 4, \\ x_{B'(1)} &= \tilde{b}_1 - \tilde{a}_{1N(1)} x_{N(1)} = 1 - \big(-\frac{1}{2}\big)4 = 3, \\ x_{N'(1)} &= x_{B(2)} = \tilde{b}_2 - \tilde{a}_{2N(1)} x_{N(1)} = 2 - \frac{1}{2}4 = 0, \\ x_{N'(2)} &= x_{N(2)} = 0. \end{aligned}$$

This gives the same solution which was obtained directly in Example 2.3 (b) by using the definition of the basic feasible solution with respect to $B' = (1, 3)$.

□

It is the idea of the simplex method to move iteratively from the basic feasible solutions to basic feasible solution until an optimal basic feasible solution is reached. Nevertheless, it still remains to be shown that there is always an optimal basic feasible solution and that the procedure is, indeed, finite. But first we show how to organize the basic exchange and the optimality test efficiently.

If we write the objective function as $-z + c_1 x_1 + \cdots, + c_n x_n = 0$, then the objective function and the constraints can be stored in a matrix which we write in tableau form, called the *starting tableau* $T = \big(t_{ij}\big)_{\substack{i \in \{0,1,\ldots,m\} \\ j \in \{0,1,\ldots,n,n+1\}}}$:

$$T = \begin{array}{|c|c|c|} \hline 1 & c_1 \ \ldots \ c_n & 0 \\ \hline 0 & a_{11} \ \ldots \ a_{1n} & b_1 \\ \vdots & \vdots & \vdots \\ 0 & a_{m1} \ \ldots \ a_{mn} & b_m \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 1 & c^T & 0 \\ \hline 0 & A & b \\ \hline \end{array}$$

Here, T represents a system of linear equations with $m+1$ equations. The $(n+1)$-st column contains the information about the right-hand sides of the equations. If B is a basis, then we denote by T_B the nonsingular $(m+1)\times(m+1)$ matrix

$$T_B := \left(\begin{array}{c|c} 1 & c_B^T \\ \hline 0 & A_B \end{array}\right).$$

It is easy to verify that

$$\begin{aligned} T_B^{-1} &:= \left(\begin{array}{c|c} 1 & -c_B^T A_B^{-1} \\ \hline 0 & A_B^{-1} \end{array}\right), \\ T_B^{-1}T &= \left(\begin{array}{c|c|c} 1 & c^T - c_B^T A_B^{-1} A & -c_B^T A_B^{-1} b \\ \hline 0 & A_B^{-1} A & A_B^{-1} b \end{array}\right) =: T(B). \end{aligned}$$

We call $T(B)$ the *simplex tableau* associated with the basis B. Since T_B^{-1} is nonsingular, $T(B)$ represents the same system of linear equations as the starting tableau T. The entries of $T(B)$ can be interpreted as follows:

(i) The first column is always the first standard vector E_1^T. It emphasizes the character of the 0^{th} row as an equation. Later on, we will omit this column.

(ii) For $j = B(i)$ we have:

$$A_B^{-1}A_j = E_i^T \qquad \text{(being a column)}.$$

Furthermore,

$$c_j - c_B^T A_B^{-1} A_j = c_j - c_j = 0 \quad \text{then}$$

$T(B)$ contains in the column corresponding to the i^{th} basic variable $x_{B(i)}$ the value 0 in row 0 and, then, the i^{th} unit vector E_i^T with m components.

(iii) For $j = N(i)$ we have:

$$A_B^{-1}A_j = (\widetilde{a}_{1j}, ..., \widetilde{a}_{mj})^T.$$

Furthermore,

$$t_{0j} \;=\; c_j - c_B^T A_B^{-1} A_j \;=\; \overline{c}_j$$

is the reduced cost of the non-basic variable x_j.

(iv) In the bast column $A_B^{-1}b$ is the vector of the basic variables with respect to B. Consequently, $c_B^T A_B^{-1} b$ is the negative of the objective value of the current basic feasible solution.

Example 2.5. (Continuation of Example 2.4). If we consider again Example 2.2 with $B = (1, 2)$, then

$$T \;=\; \begin{array}{|c|c|c|}\hline 1 & \text{-1 0 0 0} & 0 \\ \hline 0 & \text{-1 1 1 0} & 1 \\ 0 & \text{1 1 0 1} & 3 \\ \hline \end{array}$$

and because of

$$A_B^{-1} \;=\; \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \text{ and}$$

$$c_B^T A_B^{-1} \;=\; \begin{pmatrix} -1 & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} \\ \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \;=\; \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

we get

$$T_B^{-1} \;:=\; \left(\begin{array}{c|cc} 1 & -0.5 & 0.5 \\ \hline 0 & -0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{array}\right).$$

Hence, the simplex tableau corresponding to B is

$$T(B) \;=\; T_B^{-1} T \;=\; \begin{array}{|c|c|c|}\hline 1 & \text{0 0 -0.5 0.5} & 1 \\ \hline 0 & \text{1 0 -0.5 0.5} & 1 \\ 0 & \text{0 1 0.5 0.5} & 2 \\ \hline \end{array}$$

Following the interpretation of $T(B)$, the reduced costs $\overline{c}_3 = -\frac{1}{2}$, $\overline{c}_4 = 0.5$ of the non-basic variables can be taken from the 0^{th} of $T(B)$. It can be easily seen that the optimality condition is not satisfied (which we know already from Example 2.3).
Looking at the last column of the tableau, it can be seen that $x_1 = 1$, $x_2 = 2$

are the values of the basic variables in the basic feasible solution, yielding an objective value of $-t_{0n+1} = -1$.

□

If $\exists j \in \{1, ..., n\} : t_{0j} < 0$, then we try to more the non-basic variable x_{nj} into the basis. Since the entries of the tableau are

$$t_{1j} = \widetilde{a}_{1j}, ..., t_{mj} = \widetilde{a}_{mj},$$

and

$$t_{1n+1} = \widetilde{b}_1, ..., t_{mn+1} = \widetilde{b}_m,$$

we can perform the min ratio test using the *simplex tableau* in a very simple way:

$$\delta = x_j := \min \left\{ \frac{\widetilde{t}_{i\,n+1}}{\widetilde{t}_{i\,j}} \,\middle|\, \widetilde{t}_{i\,j} > 0 \right\}.$$

Thus, an unbounded objective function can be recognized by the fact that the column corresponding to one of the non - basic variables x_j with $t_{0j} < 0$ contains only entries ≤ 0 (cf. Theorem 2.2).

If $\delta = \frac{t_{r\,n+1}}{t_{r\,j}}$, a pivot operation is carried out with the element $t_{r\,j} > 0$, i.e., we transform the j^{th} column of $T(B)$ into the j^{th} standard (column) vector E_j^T using only elementary row operations. The resulting tableau is the simplex tableau $T(B')$ with respect to the new basis B'.

Example 2.6. (Continuation of Example 2.5). Let us come back to Example 2.5:
Since $t_{03} = -\frac{1}{2}$, we try to move x_3 into the basis. The min ratio rule yields

$$\delta = x_3 = \frac{t_{25}}{t_{23}} = \frac{2}{0.5} = 4.$$

Hence, we pivot the last tableau (see above) with the element $t_{23} = 0.5$.

$$T(B) = \begin{array}{|c|c|c|} \hline 1 & 0\ 0\ \text{-}0.5\ 0.5 & 1 \\ \hline 0 & 1\ 0\ \text{-}0.5\ 0.5 & 1 \\ \hline 0 & 0\ 1\ 0.5\ 0.5 & 2 \\ \hline \end{array} \sim \begin{array}{|c|c|c|} \hline 1 & 0\ 1\ 0\ 1 & 3 \\ \hline 0 & 1\ 1\ 0\ 1 & 3 \\ \hline 0 & 0\ 2\ 1\ 1 & 4 \\ \hline \end{array} = T(B').$$

In $T(B')$, all reduced cost values $(= t_{0j}, j \in \{1, ..., m\}$ are ≥ 0: The corresponding basic solution with

$$x_1 = 3,\, x_3 = 4\,,\, x_2 = x_4 = 0$$

is optimal.

□

If $t_{0j} \geq 0,\ \forall j \in \{1, ..., n\}$ and $t_{in+1} \geq 0,\ \forall i \in \{\,1, ..., m\,\}$, then $T(B)$ is called an *optimal (simplex) tableau*. We summarize:

Algorithm (Simplex Method for the Solution of LPs of the type $\min c^T x : Ax = b,\ x \geq 0$)

(Input) Basic feasible solution (x_B, x_N) with respect to basis B.
(1) Compute the simples tableau $T(B)$.
(2) If $t_{0j} \geq 0\ \forall j = 1, \ldots, n$,
(STOP) (x_B, x_N) with $x_{B(i)} = t_{in+1}\ (i = 1, \ldots, m)\ x_N = 0$
and with the objective value
$-t_{0n+1}$ is an optimal solution of LP.
(3) Choose j with $t_{0j} < 0$.
(4) If $t_{ij} \leq 0\ \forall i = 1, \ldots, m$ (STOP) The LP is unbounded.
(5) Determine $r \in \{1, \ldots, m\}$ with
$$\frac{t_{rn+1}}{t_{ij}} = \min\left\{ \frac{t_{rn+1}}{t_{ij}} \;\middle|\; t_{ij} > 0 \right\}$$
and pivot with t_{rj}.
Goto (2).

For the well-definiteness of (1) we state:

Theorem 2.3 (Fundamental Theorem of *LPs*). *Suppose that an LP*

$$(\mathrm{LP})_{st} \qquad \left\{ \begin{array}{lrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{array} \right.$$

is given and that $P := \{\, x \in \mathbb{R}^n \,|\, Ax = b,\, x \geq 0 \,\} \neq \emptyset$. *Then, there exists a basic feasible solution.*

Proof. cf., e.g., Hamacher, Klamroth, (2000). □

We conclude our first chapter by an example:

Example 2.7. Solve the *LP*

$$(\text{LP}) \qquad \begin{cases} \text{maximize} & 3x_1 & + & x_2 & & \\ \text{subject to} & -x_1 & + & x_2 & \leq & 1 \\ & x_1 & - & x_2 & \leq & 1 \\ & & & x_2 & \leq & 2 \\ & x_1 & , & x_2 & \geq & 0 \end{cases}$$

by the Simplex Method.

Solution:
First we transform $\mathcal{LP}$ into standard form:

$$(\text{LP})_{st} \qquad \begin{cases} \text{(-) minimize} & -3x_1 & - & x_2 & & & & & & & \\ \text{subject to} & -x_1 & + & x_2 & +x_3 & & & & & = & 1 \\ & x_1 & - & x_2 & & + & x_4 & & & = & 1 \\ & & & x_2 & & & & + & x_5 & = & 2 \\ & x_1 & , & x_2 & , & x_3 & , x_4 & , x_5 & \geq & 0 \end{cases}$$

□

(The "-" in front of "min" will be omitted in the following.) As a consequence, the objective function value of the current basic feasible solution is $-(-t_{06}) = t_{06}$ in the following tableaus (and not $-t_{06}$).

A basic feasible starting solution is given by $B = (3, 4, 5)$, i.e., the slack variables are the basic variables. The first simplex tableau is therefore (without the O^{th} column):

$$T(B) = \begin{array}{|c|c|} \hline \text{-3 -1 0 0 0} & 0 \\ \hline \text{-1 1 1 0 0} & 1 \\ \textbf{1} \text{ -1 0 1 0} & 1 \\ \text{0 1 0 0 1} & 2 \\ \hline \end{array}$$

While applying the Simplex Method, we indicate the pivot column (i.e., the column corresponding to a non-basic variable x_j with $t_{0j} < 0$) and the pivot row (i.e., the row r determined by (5)) by representing the pivot element in boldface.

$$T(B) \sim \begin{array}{|ccccc|c|}\hline 0 & -4 & 0 & 3 & 0 & 3 \\ \hline 0 & 0 & 1 & 1 & 0 & 2 \\ 1 & -1 & 0 & 1 & 0 & 1 \\ 0 & \mathbf{1} & 0 & 0 & 1 & 2 \\ \hline \end{array} \sim \begin{array}{|ccccc|c|}\hline 0 & 0 & 0 & 3 & 4 & 11 \\ \hline 0 & 0 & 1 & 1 & 0 & 2 \\ 1 & 0 & 0 & 1 & 1 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ \hline \end{array} \quad \text{optimal tableau}$$

(STOP) (x_B, x_N) with

$$x_B = \begin{pmatrix} x_3 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix} \quad \text{and} \quad x_N = \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

hence,

$$x^* := \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

is optimal with objective value 11.

□

With our next chapter we remain in the field of LP, but follow a basically different approach in solving the linear program.

Chapter 3

Linear Programming: Interior-Point Methods

3.1 Introduction

In the 80s of 20th century, it was discovered that many large linear programs could be solved efficiently by formulating them as nonlinear problems and solving them with various modifications of nonlinear algorithms such as Newton's method. One characteristic of these methods was that they required to satisfy inequality constraints strictly. So, they soon became called *interior-point methods*. By the early 90s of last century, one class —so-called *primal-dual methods*— had distinguished itself as the most efficient practical approach and proved to be a strong competitor to the simplex method on large problems. These methods are the focus of this chapter.

The simplex method can be quite inefficient on certain problems: The time required to solve a linear program may be exponential in the problem's size, as measured by the number of unknowns and the amount of storage needed for the problem data. In practice, the simplex method is much more efficient than this bound would suggest, but its poor worst-case complexity motivated the development of new algorithms with better guaranteed performance. Among them is the ellipsoid method proposed by Khachiyan (1979), which finds a solution in time that is at worst polynomial in the problem size. Unfortunately, this method approaches its worst-case bound on all problems as is not competitive with the simplex method. Karmarkar (1984) announced a projective algorithm which also has polynomial complexity, but it came with the added inducement of a good practical behaviour. The initial claims of excellent performance on large linear programs were never fully borne out,

but the announcement prompted a great deal of research activity and a wide array of methods described by such labels as "affine-scaling ", "logarithmic-barrier", "potential-reduction", "path-following", "primal-dual", and "infeasible interior point". Many of the approaches can be motivated and described independently of the earlier ones of Karmarkar or of so-called log-barrier, etc..

Interior-point methods share common features that distinguish them from the simplex method. Each interior-point iteration is expensive to compute and can make a significant progress towards the solution, while the simplex method usually requires a large number of inexpensive iterations. The simplex method works its way around the boundary of the feasible polytope, testing a sequence of vertices in turn until it finds the optimal one. Interior-point methods approach the boundary of the feasible set in the limit. They may approach the solution either from the interior or from the exterior of the feasible region, but they never actually lie on the boundary of this region.

3.2 Primal-Dual Methods

Outline: Let our LP be given in standard form:

$$(\mathrm{LP})_{st} \qquad \begin{cases} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0; \end{cases}$$

cf. Chapter 2. The dual (linear) problem, LDP, is defined by

$$(\mathrm{DP}) \qquad \begin{cases} \text{maximize} & b^T y & & \\ \text{subject to} & A^T y + s & = & c, \\ & s & \geq & 0, \end{cases}$$

where $y \in \mathbb{R}^m$ (dual variable) and $s \in \mathbb{R}^n$ (slack variable). "Primal-dual" solutions of $(\mathrm{LP})_{st}$, (DP) fulfill the (first-order) necessary optimality conditions from general optimization theory which are called *Karush-Kuhn-Tucker conditions* (cf. Section 1.1). We state them here as follows (exercise):

$$(\mathrm{NOC}) \qquad \begin{cases} A^T y + s & = & c, & \\ Ax & = & b, & \\ x_j s_j & = & 0 & \forall j \in \{1, 2, \ldots, n\}, \\ (x, s) & \geq & 0.^1 & \end{cases}$$

[1] Please consider the following Lagrange function: $L(x, y, s) := c^T x - y^T(Ax - b) - s^T x$. The condition $(x, s) \geq 0$ just means nonnegativity of all x_j, s_j.

Primal-dual methods find solutions (x^*, y^*, s^*) of (NOC) by applying variants of Newton's method to the three groups of equalities, and modifying the search directions and step lengths so that the inequalities $(x, s) \geq 0$ are satisfied strictly at every iteration. The equations in (NOC) are only mildly nonlinear and so are not difficult to solve by themselves. However, the problem becomes more difficult when we add the nonnegativity requirement $(x, s) \geq 0$. The nonnegativity condition is the source of all the complications in the design and analysis of interior-point methods.

To derive primal-dual interior-point methods, we restate (NOC) in a slightly different form by a mapping $F : \mathbb{R}^{2n+m} \longrightarrow \mathbb{R}^{2n+m}$:

$$\text{(NOC)} \qquad \begin{cases} F(x, y, s) := \begin{pmatrix} A^T y + s - c \\ Ax - b \\ XS\,\mathbf{1} \end{pmatrix} & \stackrel{!}{=} \quad 0, \\ (x, s) & \geq \quad 0, \end{cases}$$

where

$$X := \operatorname{diag}(x_1, x_2, \ldots, x_n),$$

$$S := \operatorname{diag}(s_1, s_2, \ldots, s_n), \text{and}$$

$$\mathbf{1} := (1, 1, \ldots, 1)^T.$$

Primal-dual methods generate iterates (x^k, y^k, s^k) that satisfy the bounds $(x, s) \geq 0$ strictly, i.e., $x^k > 0$ and $s^k > 0$. By respecting these bounds, the interior-point methods avoid spurious solutions, i.e., points that satisfy $F(x, y, s) = 0$ but not $(x, s) \geq 0$. Spurious solutions abound and do not provide useful information about solutions of $(\text{LP})_{st}$ and (DP). So, it is reasonable to exclude them altogether from the region of search. Many interior-point methods actually require (x^k, y^k, s^k) to be strictly feasible, i.e., to satisfy the linear equality constraints for the primal and dual problems. We put

$$\mathcal{M} := \left\{(x, y, s) \;\middle|\; Ax = b, A^T y + s = c, (x, s) \geq 0\right\},$$

$$\mathcal{M}^0 := \left\{(x, y, s) \;\middle|\; Ax = b, A^T y + s = c, (x, s) > 0\right\};$$

so, $\mathcal{M}^0$ is some relative interior of $\mathcal{M}$. Herewith, the strict feasibility condition can be written concisely as

$$(x^k, y^k, s^k) \in \mathcal{M}^0 \quad \forall k \in \mathbb{N}_0. \tag{SF}$$

Primal-dual interior point methods consist of both

(a) a procedure for determining the step,

(b) a measure of desirability of each point in search space.

The search direction procedure has its origins in Newton's method, here, applied to nonlinear equations $F \stackrel{!}{=} 0$ in (NOC). Newton's method forms a linear model for F around the current point and obtains the search direction $(\Delta x, \Delta y, \Delta s)$ by solving the following system of linear equations (exercise):

$$DF(x, y, s) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = -F(x, y, s).$$

If $(x, y, s) \in \mathcal{M}^0$ (strict feasibility!), then the Newton step equations become

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XS\,\mathbf{1} \end{pmatrix}. \tag{ASD}_{sf}$$

Usually, a full step along this direction is not permissible, since it would violate the bound $(x.s) \geq 0$. To avoid this difficulty, we perform a line search along the Newton direction so that the new iterate is

$$(x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$$

for some line search parameter $\alpha \in (0, 1]$. Unfortunately, we often can take only a small step along the direction (i.e., $\alpha << 1$) before violating the condition $(x, s) > 0$. Hence, the pure Newton iteration (ASD), which is known as the *affine scaling direction*, often does not allow us to make much progress towards a solution. Primal-dual methods modify the basic Newton procedure in two ways:

(i) They bias the search direction towards the interior of the "nonnegative orthant" $(x, s) \geq 0$, so that we can move further along the direction before one of the components of (x, s) becomes negative.

(ii) They keep the components of (x, s) from moving "too close" to the boundary of the nonnegative orthant.

We consider these modifications in turn.

The Central Path (embedding our framework): The central path $\mathcal{C}$ consists of strictly feasible points $(x_\tau, y_\tau, s_\tau) \in \mathcal{C}$ parametrized by a scalar $\tau > 0$, where each point solves the following system:

$$(\text{NOC})_{sf}^{\tau} \qquad \begin{cases} A^T y + s &= c, \\ Ax &= b, \\ x_j s_j &= \tau \quad \forall j \in \{1, 2, \ldots, n\}, \\ (x, s) &> 0. \end{cases}$$

From $(\text{NOC})_{sf}^{\tau}$ we can define the central path as $\mathcal{C} := \left\{(x_\tau, y_\tau, s_\tau) \;\middle|\; \tau > 0\right\}$. In fact, it can be shown that (x_τ, y_τ, s_τ) is uniquely defined for each $\tau > 0$ if and only if $\mathcal{M}^0 \neq \emptyset$. A plot of $\mathcal{C}$ for a typical problem, projected into the space of primal variables x, is shown in Figure 3.1:

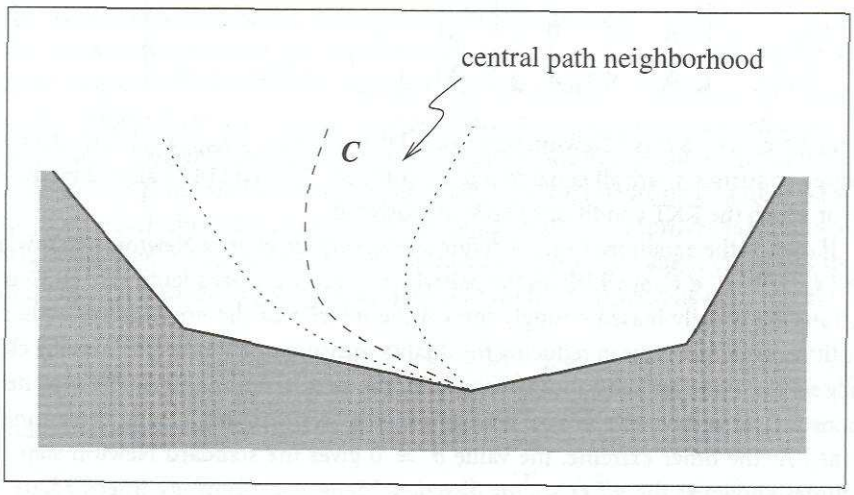


Figure 3.1: Central path, projected into x-space, showing a typical neighbourhood $\mathcal{N}$.

Another way of defining $\mathcal{C}$ is to use our mapping F (see (NOC)) and write

$$(\text{NOC})^{\tau}_{sf} \qquad F(x_\tau, y_\tau, s_\tau) = \begin{pmatrix} 0 \\ 0 \\ \tau \mathbf{1} \end{pmatrix}, \quad (x_\tau, s_\tau) > 0.$$

The equations $(\text{NOC})^{\tau}_{sf}$ approximate the ones in (NOC) more and more as $\tau \to 0$. If $\mathcal{C}$ converges to anything as $\tau \to 0$, then it must converge to a primal-dual solution of the linear problem. The central path thus guides us to a solution along a route that steers clear of spurious solutions by keeping all x- and s- components strictly positive and decreasing the pairwise products $x_j s_j$ to 0 at roughly the same rate.

Primal-dual algorithms take Newton steps towards points on $\mathcal{C}$ for which $\tau > 0$, rather that pure Newton steps for F. Since these steps are biased towards the interior of the nonnegative orthant $(x, s) \geq 0$, it usually is possible to take longer steps along them than along the pure Newton steps for F, before violating the positivity condition.

To describe the biased search direction, we introduce a centering parameter $\sigma \in [0, 1]$ and a duality measure μ defined by the average

$$\mu := \frac{1}{n} \sum_{j=1}^{n} x_j s_j = \frac{1}{n} x^T s.$$

By writing $\tau := \sigma\mu$ and applying Newton's method to $(\text{NOC})^{\tau}_{sf}$, we obtain

$$(\text{ASD})^{\sigma\mu}_{sf} \qquad \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XS\,\mathbf{1} + \sigma\mu \mathbf{1} \end{pmatrix}.$$

The step $(\Delta x, \Delta y, \Delta s)$ is a Newton step towards the point $(x_{\sigma\mu}, y_{\sigma\mu}, s_{\sigma\mu}) \in \mathcal{C}$, at which the pairwise products $x_j s_j$ are equals to $\sigma\mu$.

If $\sigma = 1$, then $(\text{ASD})^{\tau}_{sf}$ defines a centering direction, a Newton step towards the point $(x_\mu, y_\mu, s_\mu) \in \mathcal{C}$, at which $x_j s_j = \mu$ for all j. Centering directions are usually biased strongly towards the interior of the nonnegative orthant and make little progress in reducing the duality measure μ. By moving closer to $\mathcal{C}$, however, they set the scene for substantial progress on the next iteration.

If $\sigma = 0$, then we get the standard Newton step $(\text{ASD})_{sf}$. many algorithms use intermediate values of $\sigma \in (0, 1)$ to trade off between the twin

goals of reducing μ and improving centrality.

A Primal-Dual Framework: With our concepts from above, we can define a general framework for primal-dual algorithms:

Framework IPM1 Primal-Dual

Given $(x^0, y^0, s^0) \in \mathcal{M}^0$

for $k = 0, 1, 2, \ldots$

Solve

$$(\text{ASD})_{sf}^{\sigma_k \mu_k} \qquad \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k S^k \mathbf{1} + \sigma_k \mu_k \mathbf{1} \end{pmatrix},$$

where $\sigma_k \in [0, 1]$ and $\mu_k = (x^k)^T s^k / n$;

Set

$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k (\Delta x^k, \Delta y^k, \Delta s^k)$,

choosing α_k such that $(x^{k+1}, s^{k+1}) > 0$.

end(for).

The choices of centering parameter σ_k and step length α_k are crucial to the performance of the method. Techniques for controlling these parameters give rise to a wide variety of methods with varying theoretical properties.

So far, we have assumed that the starting point (x^0, y^0, s^0) is strictly feasible and, in particular, that it satisfies the linear equations $Ax^0 = b$, $A^T y^0 + s^0 = c$. All subsequent iterates also respect these constraints, because of the zero right-hand-side terms in $(\text{ASD})_{sf}^{\sigma_k \mu_k}$. However, for most problems, a strictly feasible starting point is difficult to find! Infeasible-interior-point methods require only that the components of x^0 and s^0 be strictly positive. The search direction needs to be modified so that it improves feasibility as well as centrality at each iteration, but this requirement entails only a slight change to the step equation $(\text{ASD})_{sf}^{\sigma\mu}$.

If we define the residuals for the two linear equations as

$$r_b := Ax - b, \qquad r_c := A^T y + s - c,$$

then the modified step equation is

$$(\text{ASD})_{inf}^{\sigma\mu} \qquad \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} -r_c \\ -r_b \\ -XS\,\mathbf{1} + \sigma\mu \mathbf{1} \end{pmatrix}.$$

The search direction is still a Newton step towards the point $(x_{\sigma\mu}, y_{\sigma\mu}, s_{\sigma\mu}) \in \mathcal{C}$. It tries to correct all the infeasibility in the equality constraints in a single step. If a full step is taken at any iteration (i.e., $\exists k : \alpha_k = 1$), the residuals r_b and r_c become 0, and all subsequent iterates remain strictly feasible.

Path-Following Methods: These algorithms explicitly restrict the iterates to a neighbourhood of the central path $\mathcal{C}$ and follow $\mathcal{C}$ to a solution of our LP. By preventing the iterates from coming too close to the boundary of the nonnegative orthant, they ensure that search directions calculated from each iterate make at least some minimal amount of progress towards the solution.

In path-following algorithms, the duality measure μ fills the role (to be satisfied by any optimization algorithm) of (b) introduced above. The duality measure μ_k is forced to 0, $\mu_k \to 0$ $(k \to \infty)$, so the iterates (x^k, y^k, s^k) come closer and closer to satisfy the Karsh-Kuhn-Tucker conditions (NOC)!

There are two most interesting neighbourhoods of $\mathcal{C}$:

$$(\alpha) \qquad \mathcal{N}_2(\theta) := \left\{(x, y, s) \in \mathcal{M}^0 \;\middle|\; \| XS\,\mathbf{1} - \mu\mathbf{1} \|_2 \leq \theta\mu\right\} :$$

2-norm-neighbourhood, of some $\theta \in [0, 1)$, $\| \cdot \|_2$: Euclidean norm,

$$(\beta) \qquad \mathcal{N}_{-\infty}(\gamma) := \left\{(x, y, s) \in \mathcal{M}^0 \;\middle|\; \max_{j \in \{1,2,\ldots,n\}} x_j s_j \geq \gamma\mu\right\} :$$

one-sided ∞- (or max-) norm-neighbourhood, of some $\gamma \in (0, 1]$. Typical values are $\theta = 0.5$ and $\gamma = \frac{1}{1{,}000}$. If a point lies in $\mathcal{N}_{-\infty}(\gamma)$, each $x_j s_j$ must be at least some small multiple γ of their average value μ. This requirement is quite modest, and we can make $\mathcal{N}_{-\infty}(\gamma)$ encompass most of the feasible region $\mathcal{M}$ by choosing γ close to 0. The neighbourhood $\mathcal{N}_2(\theta)$ is more restrictive, since certain points in $\mathcal{M}^0$ do not belong to $\mathcal{N}_2(\theta)$, no matter how close θ is chosen to its upper bound of 1.

The projection of neighbourhood $\mathcal{N}$ onto the x-space for a typical problem is shown as the region between the dotted lines in Figure 3.1.

By keeping all iterates inside one or another of these neighbourhoods, path-following methods reduce all the $x_j s_j$ to 0 at more or less the same rate. Our following so-called *long-step path-following algorithm* can make rapid progress because of its use of wide neighbourhood $\mathcal{N}_{-\infty}(\gamma)$, for $\gamma \approx 0$. It depends on two parameters σ_{min} and σ_{max}, which are upper and lower

bounds on the centering parameter σ_k. As usual, the search direction is obtained by solving $(\text{ASD})_{sf}^{\sigma_k\mu_k}$, and we choose the step length α_k to be as large as possible, subject to staying inside $\mathcal{N}_{-\infty}(\gamma)$.

Here and later on, we use the notation

$$\begin{array}{rcl}(x^k(\alpha), y^k(\alpha), s^k(\alpha)) & := & (x^k, y^k, s^k) + \alpha(\Delta x^k, \Delta y^k, \Delta s^k), \\ \mu^k(\alpha) & := & \frac{1}{n} x^k(\alpha) s^k(\alpha)\end{array}$$

(cf. our corresponding definitions from above).

Algorithm IPM2 (Long-Step Path-Following)

Given $\gamma, \sigma_{min}, \sigma_{max}$ with $\gamma \in (0,1)$, $0 < \sigma_{min} < \sigma_{max} < 1$,
and $(x^0, y^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$;
for $k = 0, 1, 2, \ldots$
Choose $\sigma_k \in [\sigma_{min}, \sigma_{max}]$;
Solve $(\text{ASD})_{sf}^{\sigma_k\mu_k}$ to obtain $(\Delta x^k, \Delta y^k, \Delta s^k)$;
Choose α_k as the largest value of α in $[0,1]$ such that

$$(x^k(\alpha), y^k(\alpha), s^k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma);$$

Set $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k(\alpha_k), y^k(\alpha_k), s^k(\alpha_k))$;
end(for).

The typical behaviour of our algorithm is illustrated in Figure 3.2 for the case of $n = 2$. The horizontal and vertical axes in this figure represent the pairwise product $x_1 s_1$ and $x_2 s_2$; so the central path $\mathcal{C}$ is the line emanating from the origin at an angle of 45^o. (Here, a point at the origin is a primal-dual solution if it also satisfies the feasibility conditions from (NOC) different from $x_j s_j = 0$.) In the usual geometry of Figure 3.2, the search directions $(\Delta x^k, \Delta y^k, \Delta s^k)$ transform to curves rather than lines. As Figure 3.2 shows, the bound σ_{min} ensures that each search direction stands out by moving away from the boundary of $\mathcal{N}_{-\infty}(\gamma)$ and into the relative interior of this neighbourhood. That is, small steps along the search direction improve the centrality.

Larger values of α take us outside the neighbourhood again, since the error in approximating the nonlinear system $(\text{NOC})_{sf}^{\tau}$ by the linear step equations $(\text{ASD})_{sf}^{\sigma\mu}$ becomes more pronounced as α increases. Still, we are guaranteed that a certain minimum step can be taken before we reach the boundary of $\mathcal{N}_{-\infty}(\gamma)$. A complete analysis of Algorithm IPM2 can be found in Nocedal,

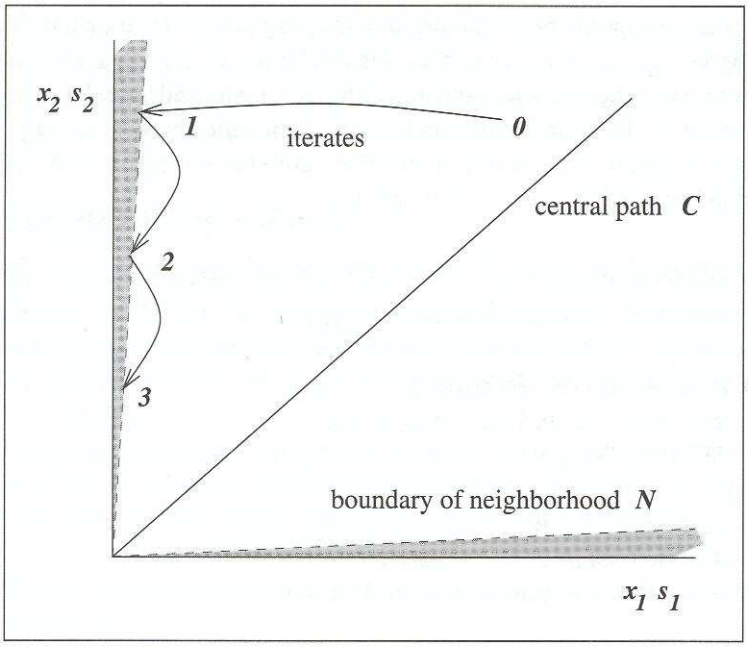


Figure 3.2: Iterates of Algorithm IPM2 plotted in (x, s)-space.

Wright (1999). It shows that the primal-dual methods can be understood without recourse to profound mathematics. This algorithm is fairly efficient in practice, but with a few more changes it becames the basis of a truly competitive method.

An infeasible-interior-point variant of Algorithm IPM2 can be constructed by generalizing the definition of $\mathcal{N}_{-\infty}(\gamma)$ to allow violation of the feasibility conditions. In this extended neighbourhood, the residual norms $\| r_b \|$ and $\| r_c \|$ are bounded by a constant multiple of the duality measure μ. By squeezing μ to 0, we also force r_b and r_c to 0, so that the iterates approach complementarity and feasibility simultaneously.

3.3 A Practical Primal-Dual Algorithm

Most existing interior-point codes are based on a predictor-corrector algorithm proposed by Mehrotra (1992). The two key features of this algorithm are:

(I) addition of a corrector step to the search direction of Framework IPM1,

so that the algorithm more closely follows a trajectory to the primal-dual solution set;

(II) adaptive choice of the centring parameter σ.

In (I), we shift the central path $\mathcal{C}$ so that it starts at our current iterate (x, y, s) and ends at the set Ω of solution points. Let us denote this modified trajectory by $\mathcal{H}$ and parametrize it by the parameter $\tau \in [0, 1]$, so that

$$\mathcal{H} = \left\{ (\widehat{x}(\tau), \widehat{y}(\tau), \widehat{s}(\tau)) \;\middle|\; \tau \in [0, 1) \right\},$$

with $\widehat{x}(0), \widehat{y}(0), \widehat{s}(0)) = (x, y, s)$ and

$$\lim_{\tau \to 1^-} (\widehat{x}(\tau), \widehat{y}(\tau), \widehat{s}(\tau)) \in \Omega$$

(see Figure 3.3).

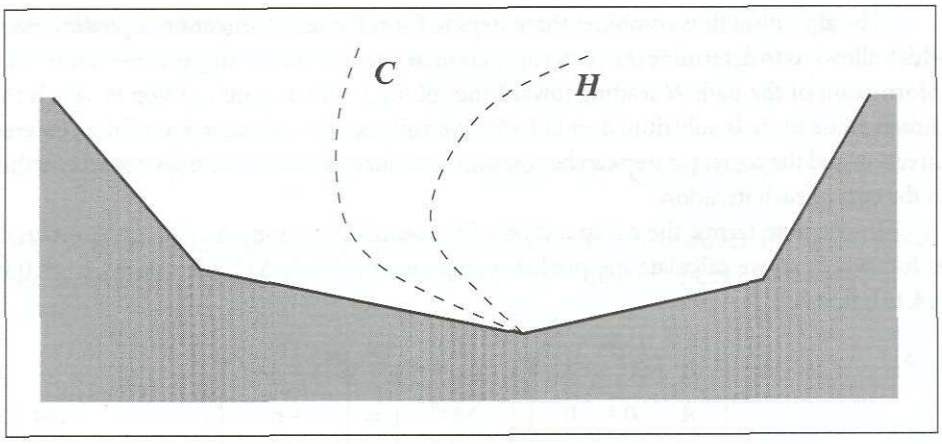


Figure 3.3: Central path $\mathcal{C}$, and a trajectory $\mathcal{H}$ from the current (noncentral) point (x, y, s) to the solution set Ω.

Algorithms from Framework IPM1 are first-order methods, in that they find the tangent to a trajectory like $\mathcal{H}$ and perform line search along it. This tangent is knows as the *predictor step*. Mehrotra's algorithm takes the next step of calculating the curvature of $\mathcal{H}$ at the current point, thereby obtaining a second-order approximation of this trajectory. The curvature is used to define the corrector step. It can be obtained at a low marginal cost, since it reuses the matrix factors from the calculation of the predictor step.

In (II), we chose the centering parameter σ adaptively, in contrast to algorithms from Framework IPM1, which assign a value to σ_k prior to calculating the search direction. At each iteration, Mehrotra's algorithm first calculates the affine-scaling direction (the predictor step) and assesses its usefulness as a search direction. If this direction yields a large reduction in μ without violating the positivity condition $(x, s) > 0$, then the algorithm concludes that little centering is needed. So, it chooses σ close to zero and calculates a centered search direction with this small value. If the affine-scaling direction is not so productive the algorithm enforces a larger amount of centering by choosing a value of σ closer to 1.

The algorithm thus combines three steps to form the research direction: a predictor step which allows us to determine the centering parameter σ_k, a corrector step using second-order information of the path $\mathcal{H}$ leading towards the solution, and centering step in which the chosen value of σ_k is substituted in $(\text{ASD})^{\sigma\mu}_{inf}$. Hereby, the computation of the centered direction and the corrector step can be combined, so that adaptive centering does not add further to the cost of each iteration.

For the computation of the search direction $(\Delta x, \Delta y, \Delta s)$ we proceed as follows:

First, we calculate the predictor step $(\Delta x^{aff}, \Delta y^{aff}, \Delta s^{aff})$ by setting $\sigma = 0$ in $(\text{ASD})^{\sigma\mu}_{inf}$, i.e.,

$$(\text{ASD})^{0,aff}_{inf} \qquad \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{aff} \\ \Delta y^{aff} \\ \Delta s^{aff} \end{pmatrix} = \begin{pmatrix} -r_c \\ -r_b \\ -XS\,\mathbf{1} \end{pmatrix}.$$

To measure the effectiveness of this direction, we find α^{pri}_{aff} and α^{dual}_{aff} to the largest length that can be taken along this direction before violating the nonnegativity conditions $(x, s) \geq 0$, and an upper bound of 1:

$$\alpha^{pri}_{aff} := \min\left\{1, \min_{j:\Delta x^{aff}_j<0} -\frac{x_j}{\Delta x^{aff}_j}\right\},$$

$$\alpha^{dual}_{aff} := \min\left\{1, \min_{j:\Delta s^{aff}_j<0} -\frac{s_j}{\Delta s^{aff}_j}\right\}.$$

Furthermore, we define μ_{aff} to be the value of μ that would be obtained by a full step to the boundary, i.e.,

$$\mu_{aff} := \frac{1}{n}\left(x + \alpha_{aff}^{pri}\Delta x^{aff}\right)\left(s + \alpha_{aff}^{pri}\Delta s^{aff}\right),$$

and set the damping parameter σ to be

$$\sigma := \left(\frac{\mu_{aff}}{\mu}\right)^3.$$

When good progress is made along the predictor direction, then we have $\mu_{aff} << \mu$, so this σ is small; and conversely.

The corrector step is obtained by replacing the right-hand-side of $(\text{ASD})_{inf}^{0,aff}$ by $(0, 0, -\Delta X^{aff}, \Delta S^{aff}\mathbf{1})$, while the centering step requires a right-hand-side of $(0, 0, \sigma\mu\mathbf{1})$. So we can obtain the complete Mehrotra step, which includes the predictor, corrector and centering step components, by adding the right-hand-sides of these three components and solving the following system:

$$(\oplus) \quad \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} -r_c \\ -r_b \\ -XS\,\mathbf{1} - \Delta X^{aff}\Delta S^{aff}\mathbf{1} + \sigma\mu\mathbf{1} \end{pmatrix}.$$

We calculate the maximum steps that can be taken along these directions before violating the nonnegativity condition $(x, s) > 0$ by formulae similar to the ones for α_{aff}^{pri} and α_{aff}^{dual}; namely,

$$\alpha_{max}^{pri} := \min\left\{1, \min_{j:\Delta x_j<0} -\frac{x_j^k}{\Delta x_j}\right\},$$

$$\alpha_{max}^{dual} := \min\left\{1, \min_{j:\Delta s_j<0} -\frac{s_j^k}{\Delta s_j}\right\},$$

and then choose the primal and dual step lengths as follows

$$\alpha_k^{pri} := \min\left\{1, \eta\alpha_{max}^{pri}\right\},$$

$$\alpha_k^{dual} := \min\left\{1, \eta\alpha_{max}^{dual}\right\},$$

where $\eta \in [0.9, 1.0)$ is chosen so that $\eta \to 1$ near the solution, to accelerate the asymptotic convergence.

We summarize this discussion by specifying Mehrotra's algorithm in the usual format.

Algorithm IPM3 (Mehrotra Predictor-Corrector Algorithm)

Given (x^0, y^0, s^0) with $(x^0, s^0) > 0$;
for $k = 0, 1, \ldots$
Set $(x, y, s) = (x^k, y^k, s^k)$ and
solve $(\text{ASD})^{0,aff}_{inf}$ for $(\Delta x^{aff}, \Delta y^{aff}, \Delta s^{aff})$;
Calculate $\alpha^{pri}_{aff}, \alpha^{dual}_{aff}$, and μ_{aff};
Set centering parameter to $\sigma = (\mu_{aff}/\mu)^3$;
Solve $(\oplus)$ for $(\Delta x, \Delta y, \Delta s)$;
Calculate α^{pri}_k and α^{dual}_k;
Set
$x^{k+1} = x^k + \alpha^{pri}_k \Delta x$,
$(y^{k+1}, s^{k+1}) = (y^k, s^k) + \alpha^{dual}_k (\Delta y, \Delta s)$;
end(for).

It is important to note that no convergence theory is available for Mehrotra's algorithm in its previous form. In fact, there are examples for which the algorithm diverges. Simple safeguards could be incorporated into the method to force it into the convergence framework of existing methods. However, most programs do not implement these safeguards, because the good practical performance of Mehrotra's algorithm makes them unnecessary.

For more information about primal-dual interior-point methods we refer to Nocedal, Wright (1999) and to Nash, Sofer (1996).

Chapter 4

Nonlinear Programming: Feasible-Point Methods

In this chapter, we examine methods which solve constrained (nonlinear) optimization problems by attempting to remain feasible at every iteration. If all the constraints are linear, maintaining feasibility is straightforward. We discuss this case first. When nonlinear constraints are present, then more elaborate procedures are required. We discuss two such approaches: sequential quadratic programming and reduced-gradient methods. Both these approaches generalize the techniques for linear constraints. Although they are motivated by the idea of maintaining feasibility at every iteration, the do not always achieve this.

4.1 Linear Equality Constraints

The majority of methods for solving problems with linear equality constraints are feasible-point methods: They start from a feasible point and move along feasible descent directions to consecutively better feasible points. There are two features making this approach particularly attractive:

(i) It is practically advantageous that all iterates are feasible. Even if the algorithm fails to solve the problem to the desired accuracy, it might still provide a feasible solution that is usable.

(ii) By restricting movement to feasible directions, the equality-constrained problem is transformed to an unconstrained one in the null space of the

constraints. This new problem may then be solved using unconstrained minimization techniques.

Let us write our problem as follows:

$$\text{(NLP)} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & Ax = b, \end{cases}$$

where f is twice-continuously differentiable: $f \in C^2$, and A is an $m \times n$ matrix of full row rank, where $m \leq n$. As in the constrained case, the methods we describe are only guaranteed to find a stationary point of the problem. In the special case where f is convex, this point will be a global minimizer of f.

Let $\overline{x}$ be a feasible point, i.e., $A\overline{x} = b$. Since any other feasible point can be reached from $\overline{x}$ by moving in a feasible direction, the solution to (NLP) can be written as $x_* = \overline{x} + p$, where p solves the problem

$$\begin{cases} \text{minimize} & f(\overline{x} + p) \\ \text{subject to} & Ap = 0, \end{cases}$$

(*Note*: $Ap = A(x_* - \overline{x}) = Ax_* - A\overline{x} = b - b = 0$).

Let Z denote an $n \times (n - m)$ basis matrix for the null-space (kernel) of A. Then, $p = Zv$ for some $(n - m)$-dimensional vector v. This problem is equivalent to the unconstrained problem

$$\text{minimize} \qquad \varphi(v) := f(\overline{x} + Zv).$$

So, we have reduced the problem of finding the best n-dimensional vector p to the unconstrained problem of finding the best $(n-m)$-dimensional vector v.

Conceptually, it is possible to minimize the reduced function φ using any of the unconstrained methods. In practice it is not necessary to provide an explicit expression for $\varphi(v)$. Instead, it is possible to work directly with the original variable x, using

$$(*) \qquad \nabla\varphi(v) = Z^T \nabla f(x), \quad \textit{and}$$

$$(**) \qquad \nabla^2\varphi(v) = Z^T \nabla^2 f(x) Z,$$

where $x = \overline{x} + Zv$. Let us shortly demonstrate this:

$$\varphi(v) = f(\overline{x} + Zv) = f(\overline{x}) + v^T Z^T \nabla f(\overline{x}) + \frac{1}{2} v^T Z^T \nabla^2 f(\overline{x}) Z v + \cdots .$$

Ideally, we would like to use the Newton direction; it is obtained by minimizing the quadratic approximation to $\varphi(v)$ obtained from the Taylor series. Setting the gradient of the quadratic approximation to 0 gives the following linear system in v:

$$\big(Z^T \nabla^2 f(\overline{x}) Z\big) v = -Z^T \nabla f(\overline{x}),$$

called the *reduced Newton* (or null-space) equation. Its solution is just

$$v = -\big(Z^T \nabla^2 f(\overline{x}) Z\big)^{-1} Z^T \nabla f(\overline{x}),$$

being an estimate of the best. In turn, it provides an estimate of the best p:

$$p = Zv = -Z\big(Z^T \nabla^2 f(\overline{x}) Z\big)^{-1} Z^T \nabla f(\overline{x}),$$

called the *reduced Newton direction* at $\overline{x}$. Now, we can derive the equality-constrained analog of the classical Newton method. The method sets

$$x_{k+1} := x_k + p_k,$$

where $p_k = -Z\big(Z^T \nabla^2 f(x_k) Z\big)^{-1} Z^T \nabla f(x_k)$ is the reduced Newton direction at x_k. This is just the mathematical formulation of the method, and in practice, explicit inverses are not normally computed. The method does not require that the reduced function be formed explicitly.

Example 4.1. We consider the problem:

$$\text{(NLP)}_{=} \qquad \begin{cases} \text{minimize} \quad f(x) = \frac{1}{2}x_1^2 - \frac{1}{2}x_3^2 + 4x_1x_2 + 3x_1x_3 - 2x_2x_3 \\ \text{subject to} \quad x_1 - x_2 - x_3 = -1 \\ \qquad \text{(i.e., } A := (1, -1, -1), \quad b = (-1)). \end{cases}$$

As a basis matrix for the null-space of A we choose

$$Z = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The reduced gradient at $\overline{x} := (1,1,1)^T$, being the feasible point which we consider, is

$$Z^T \nabla f(\overline{x}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 8 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 10 \\ 8 \end{pmatrix},$$

and the reduced Hessian matrix at $\overline{x}$ is

$$Z^T \nabla^2 f(\overline{x}) Z = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 3 \\ 4 & 0 & -2 \\ 3 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 9 & 6 \\ 6 & 6 \end{pmatrix}.$$

The reduced Newton equation yields

$$v = \begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix};$$

hence, the reduced Newton direction is

$$p = Zv = \begin{pmatrix} -\frac{4}{3} \\ -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix}.$$

Since the objective function is quadratic and the reduced Hessian matrix is positive definite, a step length of $\alpha = 1$ leads to the optimum $x^* = \left(-\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T$. At x^*, the reduced gradient is $Z^T \nabla f(x^*) = Z^T (2, -2, 2)^T = 0$ as expected. The corresponding Lagrange multiplier is $\lambda^* = 2$, because

$$\nabla f(x^*) = 2A^T.$$

□

The reduced Newton direction is invariant with respect to the null-space matrix Z: Any choice of the basis matrix Z will yield the same search direction p. Numerically, however, the choice of Z can have dramatic effect on the computation.

The classical reduced Newton method has all the properties of the classical Newton method. In particular, if the reduced Hessian matrix at the solution is positive definite, and if the starting point is sufficiently close to

the solution, then the iterates will converge quadratically. In the more general case, however, the method may diverge or fail. Then, some globalization strategy should be used.

If x_k is not a local solution and if the reduced Hessian matrix is positive definite, then the reduced Newton direction is a descent direction, since

$$\begin{aligned} p^T\nabla f(x_k) &= -\nabla f(x_k)Z\big(Z^T\nabla^2 f(x_k)Z\big)^{-1}Z^T\nabla f(x_k), \\ &< 0. \end{aligned}$$

If the Hessian matrix is not positive definite, then the search direction may not be a descent direction, and worse still, it may not be defined. Then, the modified factorizations can be applied to the reduced Hessian matrix to provide a descent direction; see Nash, Sofer (1996).

Other compromises on Newton's method may be made to obtain cheaper iterations. The simplest of all methods is of course the *steepest-descent method*. For the reduced function, this strategy gives the direction

$$v = -Z^T\nabla f(x_k)$$

in the reduced space, which yields the reduced steepest-descent direction

$$p = -ZZ^T\nabla f(x_k)$$

in the original space. Here, Z may be any null-space matrix for A. However, the direction will vary with the particular choice of Z, unlike the reduced Newton direction!

Example 4.2. The reduced gradient at the initial point of Example 4.1 is $Z^T\nabla f(\overline{x}) = (10, 8)^T$; hence, the reduced steepest-descent direction is $p = -ZZ^T\nabla f(\overline{x}) = (-18, -10, -8)^T$. Had we chosen

$$\widehat{Z} = \begin{pmatrix} 2 & 0 \\ 1 & 4 \\ 1 & -4 \end{pmatrix}$$

as the null-space matrix for A, the reduced gradient would be $\widehat{Z}^T\nabla f(\overline{x}) = \widehat{Z}^T(8, 2, 0)^T = (18, 8)^T$, and the reduced steepest-descent direction would be $p = -\widehat{Z}\widehat{Z}^T\nabla f(\overline{x}) = (-36, -50, 14)^T$. □

The reduced steepest-descent method has the same properties as that of its classical unconstrained counterpart. The iterations are cheap, but convergence may be very slow. Quasi-Newton methods are a more sophisticated

compromise. A common approach is to construct an approximation B_k to the reduced Hessian matrix at x_k. Let Z be a basis matrix for the null-space of A. The search direction is computed as $p = Zv$, where v is obtained by solving

$$B_k v = -Z^T \nabla f(x_k).$$

The approximation B_k is updated in much the same way as in the unconstrained case, except that all quantities are in the reduced space. For example, the symmetric rank-one update formula becomes

$$B_{k+1} = B_k + \frac{(\overline{y}_k - B_k \overline{s}_k)(\overline{y}_k - B_k \overline{s}_k)^T}{(\overline{y}_k - B_k \overline{s}_k)^T \overline{s}_k},$$

where $\overline{y}_k = Z^T\big(\nabla f(x_{k+1}) - \nabla f(x_k)\big)$ and $\overline{s}_k = Z^T(x_{k+1} - x_k)$.

In principle, it is possible to solve an equality-constrained problem using any classical technique from the unconstrained case. In practice, the numerical difficulties encountered when solving equality-constrained problems are not quite the same as those encountered when solving unconstrained problems, and it is not always possible to solve a large equality-constrained problem by simple applying general-purpose software for unconstrained optimization. One reason is that the reduced Hessian matrix in a constrained problem is often different in structure from the Hessian matrix in an unconstrained minimization problem:

Example 4.3. Consider the quadratic problem

$$(\text{NLP})_= \qquad \begin{cases} \text{minimize} & f(x) = \frac{1}{2}x_1^2 + x_2^2 + 2x_3^2 + 4x_4^2 \\ \text{subject to} & x_1 + x_2 + x_3 + x_4 = 1. \end{cases}$$

Taking

$$Z = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

as a basis for the null-space of the constrained matrix $A = (1\ 1\ 1\ 1)$, we obtain

$$\nabla^2 f(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 8 \end{pmatrix}, \qquad \text{and}$$

$$Z^T \nabla^2 f(x) Z = \begin{pmatrix} \frac{15}{4} & \frac{3}{4} & -\frac{5}{4} \\ \frac{3}{4} & \frac{15}{4} & -\frac{9}{4} \\ -\frac{5}{4} & -\frac{9}{4} & \frac{15}{4} \end{pmatrix}.$$

Thus, the reduced matrix $Z^T \nabla^2 f(x) Z$ is dense, even though the matrix $\nabla^2 f(x)$ is sparse. The special diagonal structure of the Hessian matrix is destroyed by the reduction! □

To overcome these problems, special implementations that are tailored to the equality-constrained problem may be needed. For example, if a conjugate-gradient method is used to solve the reduced Newton equation, then it is not necessary to form the reduced Hessian matrix explicitly.

Once an optimal solution to the equality-constrained problem is obtained, the associated vector of Lagrange multipliers is computed. There are several reasons for this.

(i) The Lagrange multipliers measure the sensitivity of the solution to changes in the constraints.

(ii) The equality-constrained problem could be one of the sequence of problems generated by an algorithm for solving a problem with inequality constraints. In this case, the Lagrange multipliers indicate how to improve the current solution.

The optimality conditions for $(\text{NLP})_=$ can be used directly to derive algorithms. The conditions are (see Section 1.1):

$$(\text{NOC})_= \qquad \begin{cases} \nabla f(x) - A^T \lambda = 0, & \text{and} \\ b - Ax = 0, \end{cases}$$

where λ is the vector of Lagrange multipliers. If Newton's method is applied to this nonlinear system, then

$$x_{k+1} = x_k + p_k,$$

$$\lambda_{k+1} = \lambda_k + \nu_k,$$

where updates p_k and ν_k are the solutions to the Newton equations

$$\begin{pmatrix} \nabla^2 f(x_k) & -A^T \\ -A & 0 \end{pmatrix} \begin{pmatrix} p \\ \nu \end{pmatrix} \begin{pmatrix} A^T\lambda_k - \nabla f(x_k) \\ Ax_k - b \end{pmatrix}.$$

Some algorithms for constrained optimization work directly with this linear system, although some care must be taken to ensure that descent directions are obtained. This linear system is closely related to the reduced Newton equation derived above.

4.2 Computing the Lagrange Multipliers λ

We still consider the linear equality-constrained problem $(\text{NLP})_=$ from Section 4.1. Assume, that the linear independence constraint qualification holds, i.e., the regularity condition saying that the rows of A are linearly independent. Consider now the optimality condition

$$\nabla f(x^*) = A^T\lambda^* \qquad \text{(cf. } (\text{NOC})_=\text{)}.$$

This is a system of n equations in $m \leq n$ unknowns, so it cannot always be expected to have a solution. At most feasible points x^*, this over-determined system will be inconsistent, but if x^* is a local solution of the optimization problem, then the system will have a solution. How can such a solution λ^* be computed?

A useful tool is a matrix known as the right inverse. We define an $n \times m$ matrix A_r to be a right inverse for the $m \times n$ matrix A, if

$$AA_r = I_m.$$

It is easy to see that a matrix A has a right inverse only if it has full row rank. In this case, and if $m = n$, then the right inverse is unique, and $A_r = A^{-1}$. If $m < n$, the right inverse is generally not unique. For example, the matrices

$$\begin{pmatrix} 0.75 & 0 \\ -0.25 & 0 \\ 0 & 0.5 \\ 0 & 0.5 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

are both right inverses for the matrix

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

To see how right inverses are of use in solving the system $\nabla f(x^*) = A^T\lambda^*$, suppose that a solution to this system exists. If both sides of the equation are multiplied by A_r^T, then we obtain

$$\lambda^* = A_r^T \nabla f(x^*).$$

If the system $\nabla f(x^*) = A^T\lambda$ is consistent, its solution $\lambda^* = A_r^T\nabla f(x^*)$ is unique, even though the right inverse may not be unique. (Note, that $\nabla f(x^*) = A^T\lambda$ implies that $AA^T\lambda = A\nabla f(x^*)$, so the unique solution is $\lambda^* = (AA^T)^{-1}A\nabla f(x^*)$. In fact, if A has full row rank, then AA^T is positive definite and, hence, its inverse exists.)

The linear system $\nabla f(x^*) = A^T\lambda$ is consistent if and only if $\nabla f(x^*)$ is a linear combination of the rows of A. Hence, $\lambda^* = A_r^T\nabla f(x^*)$ will be a solution to the system if and only if

$$\big(I - AA_r^T\big)\nabla f(x^*) = 0.$$

In practice, we will almost never find a point x^* that satisfies the optimality conditions to within some specified tolerance. The point x_k will be an estimate of the optimal solution. Correspondingly, the vector $\lambda_k = A_r^T\nabla f(x_k)$ will only be an estimate of the vector of Lagrange multipliers at the solution. It is sometimes called a *first-order estimate*, because for sufficiently small ϵ, if $\| x_k - x^* \| = O(\epsilon)$ (i.e., $\frac{\| x_k - x^* \|}{\epsilon}$ is bounded under $\epsilon \to 0$) then $\| \lambda_k - \lambda^* \| = O(\epsilon)$ also.

The rest of this section is an excursion about computing a right inverse matrix.

Variable Reduction Method: Here, the variables are partitioned into m basic and $n-m$ nonbasic variables. The matrix A is partitioned into basic and nonbasic columns correspondingly. Assuming that the first m columns are basic, then,

$$A = (B \;\; N),$$

where B is an $m \times n$ nonsingular matrix, and the $n \times (n-m)$ matrix

$$Z := \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix}$$

is a basis matrix for the null space of A. The matrix

$$A_r := \begin{pmatrix} B^{-1} \\ 0 \end{pmatrix}$$

is a right-inverse matrix for A that is available with no additional computation.

Orthogonal Projection Matrix: Let the $n \times n$ matrix

$$P := I - A^T(AA^T)^{-1}A$$

be the orthogonal projection matrix into the null space of A. A right inverse for A associated with the orthogonal projection is the matrix

$$A_r := A^T(AA^T)^{-1}.$$

This matrix, which we will denote by A^+, is a special right inverse. It satisfies the following four conditions:

$$AA^+A = A, \qquad (AA^+)^T = AA^+,$$

$$A^+AA^+ = A^+, \qquad (A^+A)^T = A^+A.$$

It can be shown that, for any $m \times n$ matrix A, there is a unique $n \times m$ matrix A^+ that satisfies these conditions. We call A^+ be the *Penrose-Moore generalized inverse* of A. If A has full row rank, then,

$$A^+ = A^T(AA^T)^{-1},$$

and if A has full columns rank, then,

$$A^+ = (A^TA)^{-1}A^T.$$

Formulas for A^+ can also be developed when A does not have full row or column rank; cf. Golub, Van Loan (1989).

Given a point x_k, the vector of Lagrange multipliers estimates $(A^+)^T\nabla f(x_k)$ obtained from the Penrose-Moore generalized inverse has the appealing property that it solves the problem

$$\underset{\lambda\in\mathbb{R}^m}{\text{minimize}} \ \left\| \nabla f(x_k) - A^T\lambda \right\|_2 .$$

For this reason it is called the *least-squares Lagrange multiplier estimate* at x_k.

As the condition number of AA^T is the square of the condition number of A, the computation of $(AA^T)^{-1}$ is potentially unstable. The QR factorization provides a stable approach to computing this matrix that is practical for smaller problems.

Non-Orthogonal Projection: Let D be a positive $n\times n$ matrix. Then, the $n \times n$ projection matrix

$$P_D := I - DA^T(ADA^T)^{-1}A$$

is a null space matrix for A. A right inverse for A associated with this projection is

$$A_r := DA^T(ADA^T)^{-1}.$$

QR Factorization (known from numerical mathematics): The QR factorization represents A^T as a product of an orthogonal matrix Q and an upper triangular matrix R. Denoting the first m columns of Q by Q_1 and the last $n-m$ columns by Q_2, we have

$$A^T = QR = (Q_1\ Q_2)\begin{pmatrix} R_1 \\ 0 \end{pmatrix},$$

where R_1 is an $m \times m$ triangular matrix. The $n \times (n-m)$ matrix

$$Z := Q_2$$

is an orthogonal basis for the null space of A. The matrix

$$A_r := Q_1R_1^{-T}, \qquad \left(R_1^{-T} := (R_1^{-1})^T\right)$$

is a right inverse for A available from the QR factorization at little additional cost. In fact, this matrix need not be formed explicitly: A computation of the form $\lambda_k = A_r^T\nabla f(x_k)$ may be done by first computing $y_1 = Q_1^T\nabla f(x_k)$

and then solving the triangular system $R_1\lambda = y_1$. It is easy to show that $A_r = A^T(AA^T)^{-1}$; hence, this right-inverse is in fact the Penrose-Moore generalized inverse of A.

Example 4.4. Let us construct two right inverses for

$$A := \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

If variable reduction is used, with columns 2 and 3 of A being selected as the basic columns, then

$$B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad N = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

From these we determine that

$$Z = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_r = \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

If a QR factorization of A^T is used, then (exercise)

$$Q = \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{\sqrt{2}} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \qquad \text{and}$$

$$R = \begin{pmatrix} -\sqrt{2} & 0 \\ 0 & -\sqrt{2} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

The matrix Z consists of the last two columns of Q:

$$Z = \begin{pmatrix} -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

The right inverse is obtained from the formula

$$A_r = Q_1 R_1^{-T},$$

where Q_1 consists of the first two columns of Q:

$$Q_1 = \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

and R_1 consists of the first two rows of R:

$$R_1 = \begin{pmatrix} -\sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{pmatrix}.$$

Hence,

$$A_r = \begin{pmatrix} \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix}.$$

If

$$\nabla f(x) = (7, -7, -2, -2)^T,$$

then, for the above right inverses

$$\lambda = A_r^T \nabla f(x) = \begin{pmatrix} 7 \\ -2 \end{pmatrix}.$$

No matter which right inverse has been used, the same values of the Lagrange multipliers are obtained. □

4.3 Linear Inequality Constraints

In this section, we discuss methods for solving problems with linear inequalities. The problem is now written in the following form:

$$(\text{NLP})^A_{\geqslant} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & Ax \geq b, \end{cases}$$

where f is still a C^2 function. For simplicity, we assume that the problem has only inequality constraints. (The extension to the case with equality constraints additionally is easy, and mentioned later on).

Suppose that the point x^* is a local solution to this problem. Let $\widehat{A}$ be a matrix whose rows are the coefficients of the active constraints at x^*, and let Z be a null-space matrix for $\widehat{A}$. The first-order optimality condition of Karush-Kuhn-Tucker state that there exists a vector μ^* such that

$$(\text{NOC})_{\geq} \qquad \begin{cases} \nabla f(x^*) &= \widehat{A}^T\mu^* \qquad (\Leftrightarrow Z^T\nabla f(x^*) = 0), \\ \mu^* &\geq 0 \qquad (\text{and } Ax^* \geq b). \end{cases}$$

Problems that have inequality constraints are significantly more difficult to solve than problems in which all constraints are equations. The reason is that it is not known in advance which inequality constraints are active at the solution. If we know *a priori* the correct active set, then we could ignore the inactive constraints and minimize the objective function with all active inequalities (i.e., where "=" holds) treated as equalities. In practice, unfortunately, we do not know what the active set is.

How can we resolve this combinatorial issue? A brute force approach would be to solve the equality constrained problem for all possible selections of active constraints, and then choose the best solution. Even for a small problem, however, the number of such subproblems is enormous, and the amount of work could be prohibitive!

Active-set methods attempt to overcome this difficulty by moving sequentially from one choice of active constraints to another choice that is guaranteed to produce at least a good solution. The hope is that only a fraction of the potential subproblems will be considered.

The most commonly used active-set methods are feasible-point methods. An initial feasible point, if none is provided, can be obtained much as in linear programming; see Section 1.1 and Nash, Sofer (1996).

At each iteration of the active set method, we select a working set J^w of constraints that are assumed to be active at the optimum. We attempt to minimize f with all constraints in the working set as equalities. All other constraints are considered inactive and temporarily ignored.

In general, the working set at the current point x is a subset of the constraints that are active at x, so that x is a feasible point for the working set. There may also be constraints that are active at x but that are not included in the working set; hence, the working set is not necessarily equal

to the active set.

Example 4.5.

$$(\text{NLP})^A_{\geqslant} \qquad \begin{cases} \text{minimize} \quad f(x) & & & & & & & & \\ \text{subject to} & x_1 & + & x_2 & + & x_3 & \geq & 1 \\ & x_1 & & & & & \geq & 0 \\ & & & x_2 & & & \geq & 0 \\ & & & & & x_3 & \geq & 0. \end{cases}$$

At $x_a = (0, \frac{1}{2}, \frac{1}{2})^T$, the first two inequalities are active. If both constraints are chosen for the working set, i.e., $J^w := \{1, 2\}$, then we will attempt to minimize f on the set

$$\left\{x \in \mathbb{R}^3 \;\middle|\; x_1 + x_2 + x_3 = 1, \quad x_1 = 0\right\}.$$

If only the first constraint is selected to be in the working set, then, we will attempt to minimize f on the set

$$\left\{x \in \mathbb{R}^3 \;\middle|\; x_1 + x_2 + x_3 = 1\right\}.$$

□

The problem of minimizing f subject to the constraints defined by a working set is an equality-constrained problem. Therefore, we can use any of the techniques describe in Section 4.1 to obtain a feasible direction p. We could then use this search direction within a line search method, and find an appropriate step length α. However, in our problem $(\text{NLP})^A_{\geqslant}$ with inequality constraints, an acceptable step might lead to a point that is infeasible, i.e., a point that violates one or more of the constraints that we have ignored. Geometrically, as we move from x along p, we may encounter the boundary of some constraint for some step length $\overline{\alpha}$. The value $\overline{\alpha}$ represents the largest possible step that may be taken without violating feasibility. The step length must never exceed $\overline{\alpha}$.

It is possible, that the best acceptable step length that does not exceed $\overline{\alpha}$ is $\overline{\alpha}$ itself. This step leads to a point on the boundary of the constraint. The constraint encountered is now satisfied exactly at the new point, and it is added to the working set. With the step length determined, and any necessary adjustments made to the working set, the entire process is repeated.

It is also possible, that the best acceptable step length is 0. This is an exceptional case, and occurs when $\overline{\alpha} = 0$. If this occurs, no step is taken and a constraint is added to the working set (since $\alpha = \overline{\alpha} = 0$).

Suppose, now, that we have found a point x that minimizes f on a given working set. Then, the first-order optimality conditions for the equality-constrained problem are satisfied at x, and we can compute the Lagrange multipliers corresponding to the constraints in the working set. (The Lagrange multipliers for all other constraints are assumed to be 0.) If the Lagrange multipliers are just all nonnegative, then x is also a solution to the original inequality-constrained problem and the problem is solved. However, if some Lagrange multiplier is negative, then x is not an optimal solution. The negative multiplier indicates that the function can be decreased if we move away from the corresponding constraint into the interior of the feasible region. Hence, we can drop this constraint from the working set. Now, we have a new working set, and the process is repeated. (For simplicity, we allow only one constraint to enter or leave the working set at a time, although alternatives are possible.)

Example 4.6. Figure 4.1 illustrates a possible sequence of movements in an active-set method for minimizing a convex function f on the box

$$[0,1]^3 \quad \left(= \prod_{j=1}^{3} [0,1] \subseteq \mathbb{R}^3 \right).$$

The directions of movement in this example are arbitrary feasible descent directions and do not correspond to any specific method.

Let A be the starting point. The initial working set is chosen as the active constraints at A. It consists of the upper bound constraint on x_3, which is the single constraint that is active at A. Suppose that the first search ends with B. If B is still not a minimizer of f on $x_3 = 1$, then another search is made, starting from B; etc.. Assume, finally, that E is a local minimizer of f along the constraints $x_1 = 1$ and $x_2 = 1$. if the Lagrange multipliers corresponding to these constraints are both nonnegative, then, E is a local minimizer for the original problem, and the algorithm is terminated. □

Now, let us discuss some of the above ideas in further detail. We assume that we have a point x which is feasible for the inequality constrained problem. Denote the working set at x by $\overline{J} := J^w$. Furthermore, denote the coefficient matrix for the constraints in $\overline{J}$ by $\overline{A}$ $(= A^w)$ and the corresponding right-hand-side vector by $\overline{b}$ $(= b^w)$. Let Z be a null-space matrix for $\overline{A}$. The equality constrained problem for $\overline{J}$ reads as follows:

$$\overline{(\mathrm{NLP})}^{\overline{A}}_{\geqslant} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & \overline{A}x \;\geq\; \overline{b}. \end{cases}$$

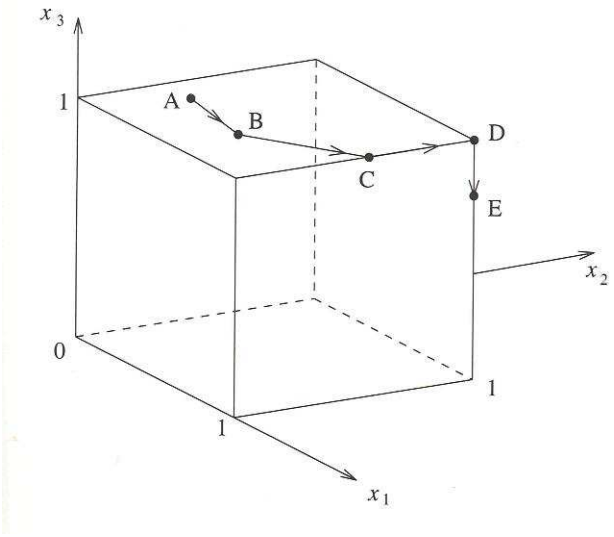


Figure 4.1: Sequence of movements in an active-set method.

This problem is commonly solved using some feasible direction method. Thus, if p is in the search direction at x, p satisfies $\overline{A}p = 0$. The step-size procedure will attempt to find an acceptable step length, while retaining feasibility with respect to all constraints.

It is easy to compute the maximum feasible step length that can be taken along a direction p using a min ratio rule (see Chapter 2):

$$\begin{aligned} \overline{\alpha} &= \max\left\{\alpha \;\middle|\; x + \alpha p \text{ is feasible } \right\} \\ &= \min\left\{ \frac{a_i^T x - b_i}{-a_i^T p} \;\middle|\; a_i^T p < 0, \quad i \notin \overline{J} \right\}. \end{aligned}$$

Now, we outlet a simple active-set method. Assume, that a feasible starting point x_0 is given and let $\overline{J}$ still be the index set of the active constraints at x_0, now. Let $\overline{A}$ be the corresponding constraint matrix, and let Z be a null-space matrix for $\overline{A}$. Set $k = 0$.

1. **Optimality Test**: If $Z^T \nabla f(x_k) = 0$, then:

   (a) If no constraints are active, then the current point is a local (unconstrained) stationary point
   —STOP.

   (b) Else, compute Lagrange multipliers:

$$\overline{\mu} = \overline{A}_r^T \nabla f(x_k).$$

(c) If $\overline{\mu} \geq 0$, then, STOP: a local stationary point has been reached. Otherwise, drop a constraint corresponding to a negative multiplier from the active set, update $\overline{J}, \overline{A}, Z$ and $\overline{A}_r$.

2. **Search Direction**: Compute a descent direction p that is feasible with respect to the constraints in $\overline{J}$.

3. **Step**: Compute a step length satisfying

$$f(x_k + \alpha p) < f(x_k)$$

and

$$\alpha \leq \overline{\alpha},$$

where $\overline{\alpha}$ is the maximum feasible step along p.

4. **Update**: Find the new point

$$x_{k+1} = x_k + \alpha p.$$

If a new constraint boundary is encountered ($\alpha = \overline{\alpha}$), then, add it to the working set and update $\overline{J}, \overline{A}, Z$ and $\overline{A}_r$ accordingly.

(if more than one constraint boundary is encountered, then, pick one of the constraints to enter the working set; this is a degenerate case.)

Set $k = k + 1$ and return to 1.

Example 4.7. Consider the problem

$$(\text{NLP})^A_{\geqslant} \qquad \begin{cases} \text{minimize} & f(x) := \frac{1}{2}(x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{subject to} & 2x_1 - x_2 \geq 0 \\ & -x_1 - x_2 \geq -4 \\ & x_2 \geq 0. \end{cases}$$

We use an active-set method to solve this problem. The equality constrained subproblems will be solved using a reduced Newton method. Since the objective function is quadratic, we use a step $\alpha = 1$ whenever it is feasible, i.e., whenever $\overline{\alpha} \geq 1$. Otherwise, we shall take the step $\alpha = \overline{\alpha}$.

Furthermore, we use the variable reduction method to compute the null-space matrix for the active constraints, and for simplicity always use the left-hand submatrix of $\overline{A}$ as the basis matrix. This gives:

$$\overline{A} = (B \;\; N), \qquad Z = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix}, \qquad A_r = \begin{pmatrix} B^{-1} \\ 0 \end{pmatrix}.$$

Let $x_0 = (0,0)^T$ be our feasible starting point. Since the constraints $j = 1$ and $j = 3$ are active, we let $\overline{J} = \{1, 3\}$, so

$$\overline{A} = \begin{pmatrix} 2 & -1 \\ 0 & 1 \end{pmatrix}, \qquad \overline{A}_r = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}.$$

The matrix Z is empty, hence the reduced gradient $Z^T \nabla f(x_0)$ vanishes trivially. We therefore compute Lagrange multipliers (for simplicity, suppressing index k)

$$\overline{\mu} = \begin{pmatrix} \overline{\mu}_1 \\ \overline{\mu}_3 \end{pmatrix} = \overline{A}_r^T \nabla f(x_0) = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} -3 \\ -4 \end{pmatrix} = \begin{pmatrix} -\frac{3}{2} \\ -\frac{11}{2} \end{pmatrix}.$$

Both multipliers are negative, thus, we should drop one of the constraints from $\overline{J}$. We drop $j = 3$, because its multiplier is more negative. Updating the working set $J := \{1\}$ gives

$$\overline{A} = (2, 1), \qquad Z = \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix}, \qquad \overline{A}_r = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix},$$

so that the new reduced gradient is $Z^T \nabla f(x_0) = -\frac{11}{2}$. We now compute the reduced Newton search direction to obtain

$$\begin{aligned} p &= -Z \left(Z^T \nabla^2 f(x_0) Z \right)^{-1} Z^T \nabla f(x_0) \\ &= -\begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} \left(\tfrac{9}{4} \right)^{-1} \left(-\tfrac{11}{2} \right) \\ &= \begin{pmatrix} \frac{11}{9} \\ \frac{22}{9} \end{pmatrix}. \end{aligned}$$

The maximum step to the boundary of the constraints is $\overline{\alpha} = \frac{12}{11}$; hence, a step length of 1 is feasible ($\alpha = 1$). The full Newton step is taken to reach the point

$$x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 1 \begin{pmatrix} \frac{11}{9} \\ \frac{22}{9} \end{pmatrix} = \begin{pmatrix} \frac{11}{9} \\ \frac{22}{9} \end{pmatrix}.$$

The next iteration begins with the optimality test:

$$Z^T \nabla f(x_1) = (\frac{1}{2}\ 1) \begin{pmatrix} -\frac{16}{9} \\ \frac{8}{9} \end{pmatrix} = 0.$$

Thus, the reduced gradient vanishes at x_1, as expected. Since a local minimum of f with respect to the working set has been found, we compute the Lagrange multiplier corresponding to the active constraint:

$$\overline{\mu} = (\mu_1) = \overline{A}_r^T \nabla f(x_1) = (\frac{1}{2}\ 0) \begin{pmatrix} -\frac{16}{9} \\ \frac{8}{9} \end{pmatrix} = -\frac{8}{9}.$$

As this multiplier is negative, we drop the constraint from $\overline{J}$. Now, we are left with $\overline{J} = \emptyset$, which means that the problem is locally unconstrained. The reduced gradient is simply the gradient itself ($Z = I$), and the search direction is simply the unconstrained Newton direction:

$$\begin{aligned} p &= -\left(\nabla^2 f(x_1)\right)^{-1} \nabla f(x_1) \\ &= -\begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} -\frac{16}{9} \\ \frac{8}{9} \end{pmatrix} \\ &= \begin{pmatrix} \frac{16}{9} \\ -\frac{4}{9} \end{pmatrix}. \end{aligned}$$

Since the largest feasible step to the boundary is

$$\overline{\alpha} = \min\left\{\frac{1}{4}, \frac{11}{2}\right\} = \frac{1}{4},$$

we use $\alpha = \frac{1}{4}$, and at the new point

$$x_2 = \begin{pmatrix} \frac{11}{9} \\ \frac{22}{9} \end{pmatrix} + \frac{1}{4} \begin{pmatrix} \frac{16}{9} \\ -\frac{4}{9} \end{pmatrix} = \begin{pmatrix} \frac{5}{3} \\ \frac{7}{3} \end{pmatrix},$$

the constraint $j = 2$ is active. We now update $\overline{J} = \{2\}$,

$$\overline{A} = (-1\ 1), \qquad Z = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \qquad \overline{A}_r = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

Again testing for optimality, we find that the reduced gradient at the new point

$$Z^T \nabla f(x_2) = (-1\ 1) \begin{pmatrix} -\frac{4}{3} \\ \frac{2}{3} \end{pmatrix} = 2.$$

Since it is not zero, we continue with a search in the Newton direction. This gives

$$\begin{aligned} p &= -Z\left(Z^T\nabla^2 f(x_2)Z\right)^{-1}Z^T\nabla f(x_2) \\ &= -\begin{pmatrix} -1 \\ 1 \end{pmatrix}(3)^{-1}2 \\ &= \begin{pmatrix} \frac{2}{3} \\ -\frac{2}{3} \end{pmatrix}. \end{aligned}$$

The maximum step to the boundary is $\overline{\alpha} = \frac{7}{2}$, hence, the step length is $\alpha = 1$. This gives

$$x_3 = \begin{pmatrix} \frac{5}{3} \\ \frac{7}{3} \end{pmatrix} + 1\begin{pmatrix} \frac{2}{3} \\ -\frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{7}{3} \\ \frac{5}{3} \end{pmatrix}.$$

At the new point, the reduced gradient is

$$Z^T\nabla f(x_3) = (-1\ 1)\begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix} = 0.$$

Since

$$\begin{aligned} \overline{\mu} &= (\overline{\mu}_2) \\ &= \overline{A}_r^T\nabla f(x_3) \\ &= (-1\ 0)\begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix} \\ &= \tfrac{2}{3} > 0, \end{aligned}$$

the point x_3 satisfies the first-order optimality conditions from $(\mathrm{NOC})_{\geqslant}$, and we terminate. Since the objective function f is strictly convex, the solution $x^* = (\frac{7}{3}, \frac{5}{3})^T$ is a strict global minimizer. The Lagrange multipliers corresponding to the three constraints are $\mu_1^* = 0, \mu_2^* = \frac{2}{3}, \mu_3^* = 0$.

The progress of the algorithm is shown in Figure 4.2.

□

One possible modification is to solve the equality constrained subproblems inexactly, whenever there is reason to believe that the working set is not the optimal active set. The rationale is that faster progress may be made by obtaining a better working set than by getting a few extra digits of accuracy on an incorrect set. This idea requires care in implementation, however. The reason is that as the solution to a problem becomes less accurate, the

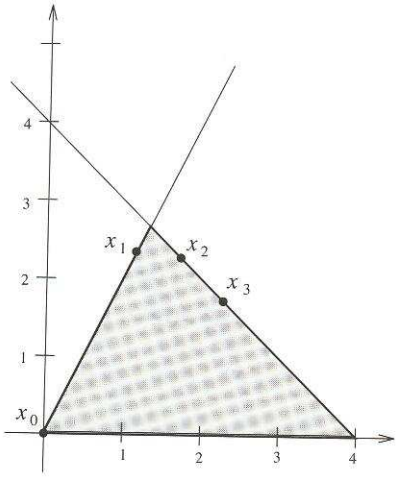


Figure 4.2: Illustration of active-set algorithm.

computed Lagrange multipliers also become less accurate. These inaccuracies can effect the sign of a computed Lagrange multiplier. Consequently, a constraint may erroneously be deleted from the working set, thereby out any potential savings.

Another possible danger is zigzagging. This phenomenon can occur if the iterated cycle repeatedly between two working sets. This situation is depicted in Figure 4.3. Zigzagging can not occur if the equality-constrained problems are solved sufficiently accurately before constraints are dropped from the working set.

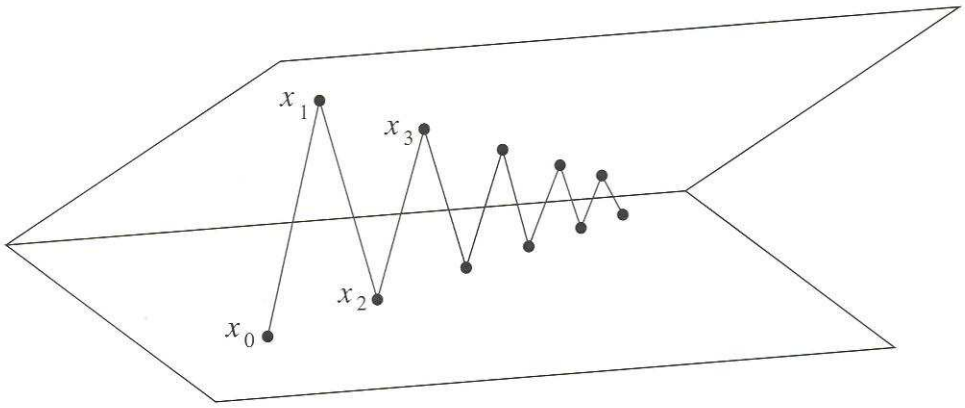


Figure 4.3: Zigzagging

Finally, we conclude by indicating how the active-set method can be adopted to solve a problem of the form

$$(\mathrm{NLP})^{A_{1,2}}_{\{\stackrel{=}{\geqslant}\}} \qquad \left\{ \begin{array}{lrcl} \text{minimize} & f(x) & & \\ \text{subject to} & A_1 x & = & b_1, \\ & A_2 x & \geq & b_2, \end{array} \right.$$

where equality constraints arise in addition to the inequality constraints. In this case, the equality constraints are kept permanently in the working set $\overline{J}$ since they must be kept satisfied at every iteration. The Lagrange multipliers for equality constraints can be positive or negative, so they do not play a role in the optimality test. The equality constraints also do not play a role in the selection of the maximum allowable step length $\overline{\alpha}$. These are the only changes that need be made to the active-set method!

At the present stage of our considerations, we could consider linear programming (cf. Chapter 2) to a great extent as special case of the methods considered in the present Chapter 4. We recall that (LP) is just $(\mathrm{NLP})_{\{\stackrel{=}{\geqslant}\}}$ with $f(x) := c^T x,\ A_1 := A,\ b_1 := b,\ A_2 := I$ and $b_2 := 0$. This is (by some hints) left as an exercise.

4.4 Sequential Quadratic Programming

Sequential quadratic programming is a popular and successful technique for solving nonlinear constrained problems. The main idea is to obtain a search direction by solving a quadratic program, i.e., a problem with a quadratic objective function and linear constraints. This approach is a generalization of Newton's method for unconstrained minimization.

Methods for solving our problem

$$(\mathrm{NLP})_{=} \qquad \left\{ \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0 \end{array} \right.$$

can be derived by applying Newton's method to the corresponding optimality conditions. Here, f and $h := (h_1, h_2, \ldots, h_m)^T$ are assumed to be of class C^2. The Lagrangian for $(\mathrm{NLP})_{=}$, arising in these Karush-Kuhn-Tucker conditions (here, Lagrange multiplier rule) is

$$L(x, \lambda) := f(x) - \lambda^T h(x),$$

and this first-order optimality condition is

$$(\text{NOC})_= \qquad \nabla L(x,\lambda) = 0.$$

Then, the formula for Newton's method is

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} p_k \\ \nu_k \end{pmatrix},$$

where p_k and ν_k are obtained as the solution to the linear system

$$\nabla^2 L(x_k,\lambda_k) \begin{pmatrix} p_k \\ \nu_k \end{pmatrix} = -\nabla L(x_k,\lambda_k).$$

This system has the form (exercise)

$$(\oplus) \qquad \begin{pmatrix} \nabla^2_{xx} L(x_k,\lambda_k) & -\nabla h(x_k) \\ \underbrace{-(\nabla h(x_k))^T}_{=:\nabla^T h(x_k)} & 0 \end{pmatrix} \begin{pmatrix} p_k \\ \nu_k \end{pmatrix} = \begin{pmatrix} -\nabla_x L(x_k,\lambda_k) \\ h(x_k) \end{pmatrix},$$

and it represents the first-order optimality conditions for the (auxiliary) optimization problem

$$(\text{QP})^k \qquad \begin{cases} \text{minimize} & \frac{1}{2} p^T \big(\nabla^2_{xx} L(x_k,\lambda_k)\big) p + p^T \big(\nabla_x L(x_k,\lambda_k)\big) \\ \text{subject to} & \big(\nabla^T h(x_k)\big) p + h(x_k) = 0, \end{cases}$$

with ν_k being the vector of Lagrange multipliers. This optimization problem is a quadratic program; i.e., it is the minimization of a quadratic function subject to linear constraints. The quadratic function is a Taylor series approximation to the Lagrangian at (x_k,λ_k), and the constraints are a linear approximation to $h(x_k+p)=0$.

In a sequential quadratic programming (SQP) method, at each iteration a quadratic problem is solved to obtain (p_k,ν_k). These are used to update (x_k,λ_k), and the process repeats at the new point. Each of the quadratic programs is solved using the technique described in Chapter 2.

Example 4.8. We apply the SQP method to the problem

$$(\text{NLP})_= \qquad \begin{cases} \text{minimize} & f(x) := e^{3x_1+4x_2} \\ \text{subject to} & h(x) := x_1^2 + x_2^2 - 1 = 0. \end{cases}$$

The solution to this problem is $x^* = (-\frac{3}{5}, -\frac{4}{5})^T$ with $\lambda^* = -\frac{5}{2}e^{-5} \approx -.016845$.

We use the initial guess $x_0 = (-7, -7)^T$ and $\lambda_0 = -.01$. At this point

$$\nabla f = e^{3x_1+4x_2} \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} .02234 \\ .02979 \end{pmatrix},$$

$$\begin{aligned} \nabla^2 f &= e^{3x_1+4x_2} \begin{pmatrix} 9 & 12 \\ 12 & 16 \end{pmatrix} = \begin{pmatrix} .06702 & .08936 \\ .08936 & .11915 \end{pmatrix}; \\ &= x_1^2 + x_2^2 - 1 = -.02, \end{aligned}$$

$$\nabla h = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = \begin{pmatrix} 1.4 \\ 1.4 \end{pmatrix}, \qquad \nabla^2 h = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix};$$

$$\nabla_x L = \nabla f - \lambda \nabla h = \begin{pmatrix} .008340 \\ .015786 \end{pmatrix},$$

$$\nabla^2_{xx} L = \nabla^2 f - \lambda \nabla^2 h = \begin{pmatrix} .08702 & .08936 \\ .08936 & .13915 \end{pmatrix}.$$

The corresponding quadratic program is

$$(\text{QP})^0 \qquad \begin{cases} \text{minimize} & \frac{1}{2} p^T \big(\nabla^2_{xx} L\big) p + p^T \big(\nabla_x L\big) \\ \text{subject to} & \big(\nabla^T h\big) p + h = 0, \end{cases}$$

Its solution can be found using

$$\begin{pmatrix} \nabla^2_{xx} L & -\nabla h \\ -\nabla^T h & 0 \end{pmatrix} \begin{pmatrix} p \\ \nu \end{pmatrix} = \begin{pmatrix} -\nabla_x L \\ h \end{pmatrix}$$

or

$$\begin{pmatrix} .08702 & .08936 & 1.4 \\ .08936 & .13915 & 1.4 \\ 1.4 & 1.4 & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \nu \end{pmatrix} = \begin{pmatrix} -.008340 \\ -.015786 \\ -.020000 \end{pmatrix}.$$

The solution of the quadratic program is

$$p_0 = \begin{pmatrix} .14196 \\ -.15624 \end{pmatrix}, \qquad \nu_0 = -.004808,$$

and the new estimates of the solutions are

$$x_1 = x_0 + p_0 = \begin{pmatrix} -.55804 \\ -.85624 \end{pmatrix},$$

$$\lambda_1 = \lambda_0 + \nu_0 = -.014808.$$

The complete iteration is given in Table 4.1. Notice the rapid convergence rate, as expected for Newton's method.

k	x_k		λ_k	$\| \nabla_x L \|$	$\| h \|$
0	-.70000	-.70000	-.010000	2×10^{-2}	2×10^{-2}
1	-.55804	-.85624	-.014808	2×10^{-3}	5×10^{-2}
2	-.60779	-.79780	-.016469	3×10^{-4}	6×10^{-3}
3	-.59988	-.80013	-.016839	8×10^{-6}	7×10^{-5}
4	-.60000	-.80000	-.016845	2×10^{-9}	3×10^{-8}
5	-.60000	-.80000	-.016845	3×10^{-16}	4×10^{-15}

Table 4.1: SQP method

□

Computing the updates p_k and ν_k by solving the quadratic program corresponding to applying Newton's method to the optimality conditions for the original problem. As a result, this method will have a quadratic convergence rate provided that $\nabla^2 L(x^*, \lambda^*)$ is nonsingular. This rapid convergence rate is observed in the example. The Hessian of the Lagrangian will be nonsingular if the regularity condition and the second-order sufficiency conditions for the original optimization problem are satisfied, i.e., if $\nabla h(x^*)$ is of full rank and if $Z^T \nabla^2_{xx} L(x^*, \lambda^*) Z$ is positive definite, where Z is a basis matrix for the null-space of $\nabla h(x^*)^T$.

For the following two reasons, the SQP method outlined above is not often used in this simple form:

(i) Convergence to a local solution is not guaranteed.

(ii) It is too expensive. The method outlined above requires the Hessians of the objective function f and the constraint functions g. Once these have been obtained, a quadratic program must be solved to determine the updates p_k and ν_k. It would be desirable to reduce these requirements for derivatives, and to reduce the number of arithmetic operations required at each iteration of the algorithm. If larger problems are being solved, it would also be desirable to reduce the storage requirements by using computational techniques that do not require matrix storage.

One way to reduce this expense is to use a quasi-Newton approximation to the Hessian of the Lagrangian. If this is done, then, second derivatives need not be computed. in addition, the quasi-Newton matrix can be maintained in the form of a factorization, and in this way the arithmetic costs of the method can also be reduced.

Choosing an update formula for the quasi-Newton approximation can be a more complicated decision than in the unconstrained case.

Let us come back to (ii): Convergence results for the SQP programming method are obtained by insisting that (x_{k+1}, λ_{k+1}) be a better estimate of the solution than (x_k, λ_k). In the unconstrained case, progress is measured in terms of the objective function. In our constrained case, progress is measured in terms of an auxiliary merit function. Usually, a merit function is in the sum of terms that include the objective function and the amount of infeasibility of the constraints. If the new point reduces the objective function and reduces infeasibility, then the value of the merit function will decrease. In many instances, however, improvements in the objective value come at the expense of feasibility, and vice versa, so the merit function must balance these two aims. One example of a merit function is the quadratic penalty function

$$\mathcal{M}(x) := f(x) + \rho \overbrace{(h(x))^T}^{=:h^T(x)} h(x) \quad \Big(= f(x) + \rho \sum_{i=1}^{m} h_i^2(x) \Big),$$

where ρ is some positive number ($\rho > 0$). The grater the value of ρ, the grater the penalty for infeasibility. This merit function is a function of x only; other examples of merit functions may be functions of both x and λ.

Example 4.9. Let us use the merit function

$$\mathcal{M}(x) := f(x) + 10 h^T(x) h(x)$$

to measure the progress of the SQP method in the previous example. At the first iteration

$$\begin{aligned} x_0 &= (-.7000, -.7000)^T, \\ f(x_0) &= .007447, \\ h(x_0) &= -.02, \\ \mathcal{M}(x_0) &= .011447, \end{aligned}$$

and

$$\begin{aligned} x_1 &= (-.55804, -.85624)^T, \\ f(x_1) &= .006102, \\ h(x_1) &= .044563, \\ \mathcal{M}(x_1) &= .025961, \end{aligned}$$

so, in terms of this merit function, the point x_1 is worse than the point x_0. At the next iteration, though,

$$\begin{aligned} x_2 &= (-.60779, -.79780)^T, \\ f(x_1) &= .006641, \\ h(x_1) &= .005890, \\ \mathcal{M}(x_1) &= .006988, \end{aligned}$$

indicating that x_2 is better than both x_1 and x_0 for this merit function. □

Ideally, the merit function would be chosen so that (x^*, λ^*) would be a local minimizer of the merit function if and only if it were a local solution of the optimization problem. If this were true then a line search with respect to the merit function could be performed:

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \alpha \begin{pmatrix} p_k \\ \nu_k \end{pmatrix},$$

where α is chosen so that

$$\mathcal{M}(x_{k+1}, \lambda_{k+1}) < \mathcal{M}(x_k, \lambda_k).$$

For this to be successful, the search direction from the quadratic program would have to be descent direction for the merit function. Unfortunately, it is rarely possible to guarantee that the local minimizers of the merit function and the local solutions of the optimization problem coincide. For the merit function $\mathcal{M}(x) := f(x) + \rho\, h^T(x)h(x)$ (quadratic penalty) some of the local minimizers of $\mathcal{M}$ approach local solutions of the constrained problem in the limit as $\rho \to \infty$. In addition, $\mathcal{M}$ may have local minima at points where $h(x) \neq 0$, i.e., at infeasible points. Other merit functions have analogous deficiencies that can limit the applicability of convergence theorems, or can complicate the development of SQP methods.

In the unconstrained case we assume that search directions are descent directions with respect to the objective function. Here, we assume that either p_k or the combined vector (p_k, ν_k) is a descent direction with respect to the merit function. A common way to guarantee this is to insist that the reduced Hessian for the quadratic program be positive definite. Come back

to the constraints. If a quasi-Newton approximation to the Hessian is used to define the quadratic program, then, positive definiteness of the reduced Hessian is often guaranteed by the choice of quasi-Newton update formula. If Newton's method is used so that the Hessian in the quadratic program is $\nabla^2_{xx} L(x_k, \lambda_k)$, then, it is necessary to test if the reduced Hessian is positive definite, and to modify if it is not. This testing is more complicated than in the unconstrained case, particularly if the quadratic program is solved via the linear system ($\oplus$). In this case, the reduced Hessian may not available. and more elaborate tests for positive definiteness must be used.

Near the solution of the constrained problem, we would normally like to take a step of $\alpha = 1$ in the line search, so that the quadratic convergence rate of Newton's method could be achieved. Hence, the merit function should be chosen so that a step of $\alpha = 1$ is guaranteed to be accepted in the limit as the solution is approached. For certain merit functions this is not true. In such cases it is possible to give examples where a step of $\alpha = 1$ is unacceptable at every iteration, no matter how close the current point is to the solution.

We will ignore many of these difficulties and only state that p_k is a descent direction for the quadratic penalty merit function. Even more, we will make the simplifying assumption that the full Hessian of the Lagrangian (or some approximation to it) is positive definite. If a quasi-Newton approximation to the Hessian is used, this assumption is not reasonable, but for Newton's method it is restrictive. The lemma stated subsequently can be used to show that

$$\lim_{k \to \infty} \nabla \mathcal{M}(x_k) = 0.$$

For large values of ρ, local solutions of the constrained problem are approximative local minimizers of $\mathcal{M}(x_k)$ to within $O(1/\rho)$, so this argument provides a rough outline of a convergence theorem.

Lemma 4.1. *Assume that (p_k, ν_k) is computed as the solution to the quadratic problem*

$$(\mathrm{QP})^k_{approx} \qquad \begin{cases} \text{minimize} & \frac{1}{2} p^T \widetilde{H} p + p^T \big(\nabla_x L(x_k, \lambda_k) \big) \\ \text{subject to} & \big(\nabla^T h(x_k) \big) p + h(x_k) = 0, \end{cases}$$

where $\widetilde{H}$ is some positive-definite approximation to $\nabla^2_{xx} L(x_k, \lambda_k)$. If $p_k \neq 0$, then

$$p_k^T \nabla \mathcal{M}(x_k) < 0$$

for all sufficiently large values of ρ, where

$$\mathcal{M}(x) := f(x) + \rho h^T(x)h(x);$$

i.e., p_k is a descent direction with respect to this merit function.

Proof. See Nash, Sofer (1996). □

Finally, let us consider the following problem with inequality constraints additionally:

$$\text{(NLP)} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0, \\ & g(x) \geq 0. \end{cases}$$

Here, we can develop an SQP method for this problem as well. Our earlier quadratic program $(\text{QP})^k$ was based on a quadratic approximation to the Lagrangian function and a linear approximation to the constraints. If the same approximation is used here, we obtain:

$$(\text{QP})^k \qquad \begin{cases} \text{minimize} & \frac{1}{2}p^T\left(\nabla^2_{xx}L(x_k,\lambda_k)\right)p + p^T\nabla_x L(x_k,\lambda_k) \\ \text{subject to} & \nabla^T h(x_k)p + h(x_k) = 0, \\ & \nabla^T g(x_k)p + g(x_k) \geq 0. \end{cases}$$

The solution $(p_k, \nu_k)$to this quadratic program provide the step to the next estimate of the solution of the original constrained problem. This problem can be solved using the active-set method discussed in Section 4.3.

4.5 Reduced-Gradient Methods

Reduced-gradient methods try to maintain feasibility at every iteration. This approach has several advantages:

(i) If each estimate of the solution is feasible, the algorithm can be stopped before it converges and the approximate solution may still be useful.

(ii) Guaranteeing convergence is simpler because progress can be measured directly using the value of the objective function, rather than with an auxiliary merit function.

The disadvantage of reduced-gradient methods is the computational expense of ensuring that nonlinear constraints remain satisfied at every iteration.

We apply a version of the reduced-gradient method to our problem

$$(\text{NLP})_{=} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0. \end{cases}$$

Our derivation begins in the same way as for SQP. The Lagrangian is

$$L(x, \lambda) := f(x) - \lambda^T h(x),$$

and, if Newton's method is applied to the first-order optimality condition,

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} p_k \\ \nu_k \end{pmatrix},$$

where p_k and ν_k are obtained as the solution to the linear system

$$\nabla^2 L(x_k, \lambda_k) \begin{pmatrix} p_k \\ \nu_k \end{pmatrix} = -\nabla L(x_k, \lambda_k).$$

This linear system has the form (cf. Section 4.4):

$$(\oplus) \qquad \begin{pmatrix} \nabla^2_{xx} L(x_k, \lambda_k) & -\nabla h(x_k) \\ -\nabla^T h(x_k) & 0 \end{pmatrix} \begin{pmatrix} p_k \\ \nu_k \end{pmatrix} = \begin{pmatrix} -\nabla_x L(x_k, \lambda_k) \\ h(x_k) \end{pmatrix}.$$

At this point, the derivations for the two methods diverge. In the reduced-gradient method, we use these formulas to derive a portion of the search direction, the portion that lies in the null-space of the constraint gradients. If Z_k is a basis matrix for the null-space of $\nabla^T h(x_k)$, and Y_k is a basis matrix for the range-space of $\nabla h(x_k)$, then

$$p_k = Z_k p_Z + Y_k p_Y,$$

where p_Z is the solution to the reduced problem

$$Z_k^T \big(\nabla^2_{xx} L(x_k, \lambda_k)\big) Z_k p_Z = -Z_k^T \nabla_x L(x_k, \lambda_k).$$

This formula determines p_Z in the reduced-gradient method. If all constraints are linear, then the formula is equivalent to the formula for the reduced

Newton method derived in Section 4.1. In that case, the matrix Z_k will be the same at every iteration, and

$$\nabla^2_{xx} L(x_k, \lambda_k) = \nabla^2 f(x_k),$$

$$Z_k^T \nabla_x L(x_k, \lambda_k) = Z_k^T \nabla f(x_k).$$

Hence, the reduced-gradient method is a generalization of the reduced Newton method for linearly constrained problems. Come back to the general nonlinear case.

The remaining portion of the search direction is determined from the condition that the new estimate of the solution must be feasible: $h(x_{k+1}) = 0$. Since $x_{k+1} = x_k + p_k$, this condition has the form

$$h(x_k + Z_k p_Z + Y_k p_Y) = 0.$$

This is a system of m nonlinear equations in the m variables p_Y. (We assume that $\nabla h(x_k)$ is a matrix of null rank.) If the constraints are linear, then $p_Y = 0$. If they are nonlinear, some auxiliary algorithm must be applied to this nonlinear system to determine p_Y. For example, Newton's method could be used.

Example 4.10. Let again be given (cf. Example 4.8):

$$(\text{NLP})_= \qquad \begin{cases} \text{minimize} & f(x) := e^{3x_1+4x_2} \\ \text{subject to} & h(x) := x_1^2 + x_2^2 - 1 = 0. \end{cases}$$

The solution being $x^* = (-\frac{3}{5}, -\frac{4}{5})^T$ with $\lambda^* = -\frac{5}{2}e^{-5} \approx -.016845$. Again, let $x_0 = (-.7, -.7)^T$, even though this point is infeasible. There, $\nabla f, \nabla^2 f, h, \nabla h$ and $\nabla^2 h$ were computed in Example 4.8. An estimate of the Lagrange multiplier is needed to determine the gradients and Hessians of the Lagrangian. We compute a multiplier estimate λ_k by solving

$$\underset{\lambda}{\text{minimize}} \parallel \nabla f(x_k) - \lambda \nabla h(x_k) \parallel_2^2 .$$

At x_0 the multiplier estimate is $\lambda_0 = -.018616$. Using this value, we obtain

$$\nabla_x L = \nabla f - \lambda \nabla h = \begin{pmatrix} .008340 \\ .015786 \end{pmatrix},$$

$$\nabla^2_{xx} L = \nabla^2 f - \lambda \nabla^2 h = \begin{pmatrix} .08702 & .08936 \\ .08936 & .13915 \end{pmatrix}.$$

We use variable-reduction to compute the null-space matrix Z_k and the range-space matrix Y_k based on

$$\nabla^T h = (2x_1, 2x_2) = (B, N).$$

Then,

$$Z_k = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix} = \begin{pmatrix} -\frac{x_2}{x_1} \\ 1 \end{pmatrix},$$

$$Y_k = \frac{1}{\sqrt{x_1^2 + x_2^2}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

At this iteration

$$Z_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

The null-space portion of the search direction is obtained by solving

$$Z_k^T \big(\nabla^2_{xx} L\big) Z_k p_k = -Z_k^T \nabla_x L,$$

or

$$(.081912) p_k = -.0074466,$$

so that $p_Z = -.090909$.

The remaining portion of the search direction, $Y_k p_Y$, is determined by solving

$$h(x_k + Z_k p_Z + Y_k p_Y) = 0.$$

using Newton's method. In this example,

$$Y_k p_Y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \gamma$$

for some unknown γ, where y_1 and y_2 are the components of Y_k. If we define $\widehat{x}_k := x_k + Z_k p_Z$, then the condition for $Y_k p_Y$ has the form

$$\varphi(\gamma) := (\widehat{x}_1 + \gamma y_1)^2 + (\widehat{x}_2 + \gamma y_2)^2 - 1 = 0.$$

Applying Newton's method to this equation gives the iteration

$$\begin{aligned} \gamma_{i+1} &= \gamma_i - \frac{\varphi(\gamma_i)}{\varphi'(\gamma_i)} \\ &= \gamma_i - \frac{(\widehat{x}_1 + \gamma_i y_1)^2 + (\widehat{x}_2 + \gamma_i y_2)^2 - 1}{2y_1(\widehat{x}_1 + \gamma_i y_1) + 2y_2(\widehat{x}_2 + \gamma_i y_2)}. \end{aligned}$$

In this example, we initialize the iteration with $\gamma_0 = 0$. Then,

$$\begin{array}{ll} \gamma_0 = 0, & \varphi(\gamma_0) = -3 \times 10^{-3}, \\ \gamma_1 = .001753157308727014, & \varphi(\gamma_1) = 3 \times 10^{-6}, \\ \gamma_2 = .001751607670563711, & \varphi(\gamma_2) = 2 \times 10^{-12}, \\ \gamma_3 = .001751607669352957, & \varphi(\gamma_0) = -2 \times 10^{-16}. \end{array}$$

The overall search direction is

$$\begin{aligned} p_0 &= Z_0 p_Z + Y_0 p_Y \\ &= \begin{pmatrix} -1 \\ 1 \end{pmatrix} (-.090909) + \begin{pmatrix} -.70711 \\ -.70711 \end{pmatrix} (.001752) \\ &= \begin{pmatrix} .089671 \\ -.092148 \end{pmatrix} \end{aligned}$$

and

$$x_1 = x_0 + p_0 = \begin{pmatrix} -.61033 \\ -.79215 \end{pmatrix}.$$

The complete iteration is given in Table 4.2. The initial guess of the solution is not feasible, but all later estimate of the solution are feasible (to 16 digits). The method converges rapidly, as expected for Newton's method. □

k	x_k		λ_k	$\| \nabla_x L \|$	$\| h \|$
0	-.70000	-.70000	-.018616	5×10^{-3}	2×10^{-2}
1	-.61033	-.79215	-.016851	4×10^{-4}	0
2	-.60001	-.79999	-.016845	3×10^{-7}	0
3	-.60000	-.80000	-.016845	1×10^{-16}	1×10^{-16}

Table 4.2: Reduced-gradient method.

The reduced-gradient method corresponds to using Newton's method in the null-space of the constraints, so it can be expected to converge quadratically in nondegenerate cases. (This quadratic convergence rate is observed

in the example above.) As before, Newton's method is not guaranteed to converge, and even if it does converge, it may converge to a maximum or stationary point and not a minimum. Hence, some globalization strategy must be employed to ensure convergence to a local solution of $(\text{NLP})_=$.

A line search can be used to guarantee convergence, as was done with the SQP method. If all the solution estimates x_k are feasible points, then the value of the quadratic penalty merit function is

$$\mathcal{M}(x_k) = f(x_k) + \rho h^T(x_k)h(x_k) = f(x_k),$$

so a line search can be performed using the objective function f itself as a merit function.

The line search for a reduced-gradient method is more complicated than in the unconstrained case. for each trial value of α, the trial point must satisfy

$$h(x_k + \alpha Z_k p_Z + Y_k p_Y) = 0.$$

Hence, p_Y depends on α, and must be computed by solving a nonlinear system of equations. For large value of α there may not be p_Y that satisfies the constraints, further complicating the line search algorithm. For these reasons, it is not entirely correct to say that the reduced-gradient method produces a search direction, since in fact the method must search along an arc defined by p_Z and α.

Concerning descent, we state:

Lemma 4.2. *Assume that the reduced-gradient method is applied to the problem* $(\text{NLP})_=$. *Let* x_k *be the* k*th estimate of the solution, with* $h(x_k) = 0$. *Let* $\nabla h(x)$ *be of full rank for all* x *in a neigbourhood of* $\mathcal{N}_k$ *of* x_k. *Also assume that the null-space portion* p_Z *of the search direction is computed from*

$$Z_k^T \big(\nabla^2_{xx} L(x_k, \lambda_k)\big) Z_k p_k = -Z_k^T \nabla_x L(x_k, \lambda_k),$$

where λ_k *is an estimate of the Lagrange multipliers, and where*

$$Z_k^T \big(\nabla^2_{xx} L(x_k, \lambda_k)\big) Z_k$$

is positive definite. Define $p_Y(\epsilon)$ *as the solution to*

$$h(x_k + \epsilon Z_k p_Z + Y_k p_Y(\epsilon)) = 0.$$

If $Z_k^T \nabla_x L(x_k, \lambda_k) \neq 0$, *then*

$$f(x_k + \epsilon Z_k p_Z + Y_k p_Y(\epsilon)) < f(x_k),$$

for all sufficiently small positive values of ϵ.

Proof. See Nash, Sofer (1996). □

Variants of the reduced-gradient method have been developed that are more flexible in that they allow some violation of the constraints. such methods might be considered a compromise between reduced-gradient and SQP methods with their different advantages and disadvantages.

Chapter 5

Nonlinear Programming: Penalty and Barrier Methods

For the feasible-point methods, considered in the previous chapter, there are some major disadvantages:

(i) As the number of constraints increases, the number of potential subproblems increases exponentially. While the hope is that the algorithm will consider only a small proportion of these subproblems, there is no known method to guarantee that this indeed will be the case.

(ii) The idea of keeping the constraints satisfied exactly, although easily achieved in the case of linear constraints, is much more difficult to accomplish in the case of nonlinear constraints, and in some cases may not be desirable.

In this chapter, we discuss a group of methods, referred to as *penalizing methods*, that remove some of these difficulties. Namely, these methods solve a constrained optimization problem by solving a sequence of unconstrained optimization problems. The hope is that in the limit, the latter solutions will converge to the solution of the constrained problem. The unconstrained problems involve an auxiliary function that incorporates the objective function or the Lagrangian function, together with penalty terms that measure violations of the constraints. The auxiliary function also includes one or more parameters that determine the relative importance of the constraints in the auxiliary function. By changing these parameters appropriately, a sequence of problems is generated where the effect of the constraints becomes increasingly pronounced. In contrast to active-set methods, the auxiliary function

takes into account all constraints, even when inequalities are present; thus the combinatorial difficulties of guessing a correct active set are avoided. Further, since they do not attempt to keep the constraints satisfied exactly, penalization techniques can be more suitable for handling nonlinear constraints.

5.1 Classical Penalty and Barrier Methods

Here, we have two groups of methods:

(α) one group imposes a penalty for violating a constraint,

(β) the other group imposes a penalty for reaching the boundary of an inequality constraint.

We call (α) *penalty methods*, and (β) *barrier methods*.

We start with a geometrical motivation. Let us consider

$$(\mathrm{P})_{con} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & x \in M, \end{cases}$$

where M is the set of feasible points. Define

$$\sigma(x) := \begin{cases} 0, & \text{if} \quad x \in M, \\ +\infty, & \text{if} \quad x \notin M, \end{cases}$$

being such to say, an *infinite penalty* for violating feasibility. Hence, the problem $(\mathrm{P})_{con}$ with its constraint can be transformed into an equivalent unconstrained problem

$$(\mathrm{P})_{uncon} \qquad \text{minimize} \qquad f(x) + \sigma(x).$$

Conceptually, if we could solve $(\mathrm{P})_{uncon}$, we would be done: a point x^* solves $(\mathrm{P})_{con}$ if and only if it solves $(\mathrm{P})_{uncon}$. Unfortunately, this is not a practical idea, since the objective function of the unconstrained minimization is not defined outside of the feasible region. Even if we were to replace the "∞" by a large number, the resulting unconstrained problem would be difficult to solve because of its discontinuities.

Instead, barrier and penalty methods solve a sequence of unconstrained subproblems that are more manageable, and that gradually approximate the problem $(\mathrm{P})_{uncon}$. This is achieved by replacing the ideal penalty σ by a continuous function which gradually approaches σ.

In barrier methods, this function (called *barrier term*) approaches σ from the interior of the feasible region. It creates a barrier that prevents the iterates from reaching the boundary of the feasible region. in penalty methods, this function (called a *penalty term*) approach σ from the exterior of the feasible region. It serves as a penalty for being infeasible.

Barrier methods generate a sequence of strictly feasible iterates which converge to a solution of the problem from the interior of the feasible region. For this reason, they are called *interior-point methods*; in Chapter 3 we studied them in the special case of linear programming! Since these methods require the interior of the feasible region to be $\neq \emptyset$, they are not more appropriate for problems without equality constraints (see, however, our discussion in the linear case, including infeasibility, in Chapter 3).

In contrast, penalty methods permit the iterates to be infeasible. A gradually increasing penalty is imposed for violation of feasibility, however. Penalty methods usually generate a sequence of points that that converge to a solution of the problem from the exterior of the feasible region. These methods are usually more convenient on problems with equality constraints.

Despite their apparent differences, barrier and penalty methods have much in common. Their convergence theories are similar, and the underlying structure of their unconstrained problems is similar. Much of the theory for barrier methods can be replicated for penalty methods and vice versa. It is common to use the generic name "penalty methods" to describe both methods, with interior penalty methods referring to (β) (barrier methods), and exterior penalty methods referring to (α) (penalty methods).

Barrier Methods:
Consider the problem

$$(\mathrm{NLP})_{\geq} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & g(x) \geq 0, \end{cases}$$

where $g := (g_1, g_2, \ldots, g_s)^T$; f and g being of class C^2. Barrier methods are strictly feasible methods, i.e., the iterates lie in the interior of the feasible region. We assume, therefore, that the feasible set M has a nonempty interior:

$M^0 \neq \emptyset$; i.e., $\exists x_0 \in \mathbb{R}^n \ : \ g_j(x_0) > 0 \ \forall j \in \{1, \ldots, s\}$. We also assume that it is possible to reach any boundary point by approaching it from the interior.

Barrier methods maintain feasibility by creating a barrier that keeps the iterates away from the boundary of the feasible region. These methods use a barrier term that approaches the infinite penalty function σ. Let $\varphi(x)$ be a function which is continuous on M^0, and that becomes unbounded as the boundary of the set is approached from its interior:

$$\varphi(x) \to \infty \quad \text{as} \quad g_j(x) \to 0^+ \text{ for some } j \in \{1, 2, \ldots, s\}\,.$$

Two examples for such a function φ:

(a) $\varphi(x) = -\sum_{j=1}^{s} \ln(g_j(x))$,

(b) $\varphi(x) = -\sum_{j=1}^{s} \frac{1}{g_j(x)}$.

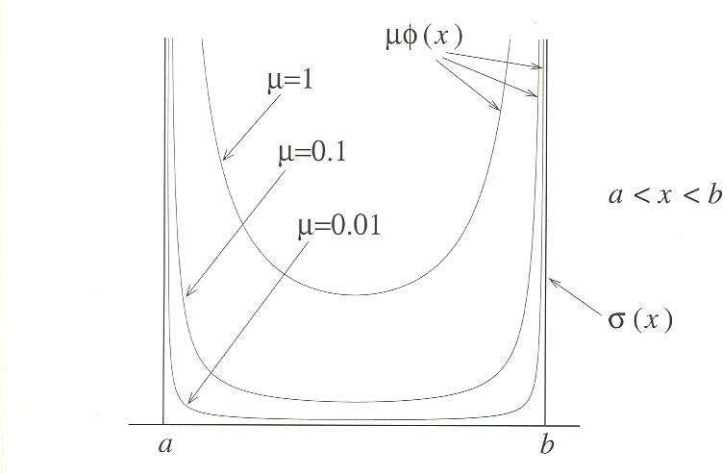


Figure 5.1: Effect of barrier term.

Now, let some $\tau > 0$ be given. Then, $\tau\varphi(x)$ will approach $\sigma(x)$ as $\tau \to 0^+$. This is demonstrated in Figure 5.1 for a one-dimensional problem with bound constraints. By adding a barrier term of the form $\tau\varphi(x)$ to the objective, we obtain a barrier function

$$\beta(x, \tau) := f(x) + \tau\varphi(x),$$

where τ is referred to as the *barrier parameter*. The best known barrier function is the *logarithmic barrier function*

$$\beta(x,\tau) := f(x) + \tau \sum_{j=1}^{s} \ln(g_j(x)),$$

but the *inverse barrier function*

$$\beta(x,\tau) := f(x) + \tau \sum_{j=1}^{s} \frac{1}{g_j(x)},$$

is also widely used. Barrier methods solve a sequence of unconstrained problems:

$$(\mathrm{NLP})^k_{uncon} \qquad \underset{x}{\text{minimize}} \;\; \beta(x,\tau_k)$$

for a sequence $\big(\tau_k\big)_{k\in\mathbb{N}_0}$ of barrier parameters $\tau_k > 0$ that decrease monotonically to zero: $\tau_{k+1} \leq \tau_k \;\forall k \in \mathbb{N}_0,\; \tau_k \to 0^+ \;(k \to \infty)$.

As the barrier term is ∞ at the boundary ∂M of the feasible set M, it acts as a repelling force which derives the minimizers of the barrier function away from the boundary into the interior M^0 of the feasible region. Thus, any minimizer of the barrier function will be strictly feasible. As the barrier parameter is decreased, however, the effect of the barrier term is diminished, so that the iterates can gradually approach the boundary ∂M.

Why do we solve a sequence of problems? It might seem better to solve a single unconstrained problem using a small μ –but this is usually not practical: When τ is small, the problems are difficult to solve ($\tau\varphi$ is close in shape to the infinite penalty σ). If τ is decreased gradually, and if the solution of one unconstrained problem is used as the starting point of the next problem, these unconstrained problems tend to be much easier to solve.

Example 5.1. Consider the nonlinear program

$$(\mathrm{NLP})_{\geq} \qquad \left\{ \begin{array}{lll} \text{minimize} & f(x) := x_1 - 2x_2 & \\ \text{subject to} & 1 + x_1 - x_2^2 \quad \geq & 0, \\ & x_2^2 \;\; \geq 0. & \end{array} \right.$$

Then, the logarithmic barrier function gives the unconstrained problems

$$(\text{NLP})^k_{uncon} \quad \underset{x}{\text{minimize}} \ \beta(x,\tau_k) := x_1 - 2x_2 - \tau_k \ln(1 + x_1 - x_2^2) - \tau_k \ln x_2$$

for a sequence of decreasing μ_k. For a specific μ_k, the first-order necessary optimality conditions are:

$$(\text{NOC})^k \qquad \left\{ \begin{array}{rcl} 1 - \dfrac{\tau_k}{1 + x_1 - x_2^2} & = & 0, \\ -2 - \dfrac{2\tau_k x_2}{1 + x_1 - x_2^2} - \dfrac{\tau_k}{x_2} & = & 0. \end{array} \right.$$

If the constraints are strictly satisfied, then, the denominators are positive. We obtain an equation for x_2:

$$x_2^2 - x_2 - \frac{1}{2}\tau = 0,$$

giving

$$x_2(\tau) := \frac{1 + \sqrt{1 + 2\tau}}{2} \qquad (\tau := \tau_k)$$

(why is the positive root taken?). Since $x_1 = x_2^2 - 1 + \tau$, we get

$$x_1(\tau) := \frac{\sqrt{1 + 2\tau} + 3\tau}{2}.$$

The unconstrained objective is strictly convex, hence, this solution is the unique local minimizer in the feasible region. As μ approaches 0 (i.e., when $k \to \infty$), we obtain

$$\lim_{\tau \to 0^+} \begin{pmatrix} x_1(\tau) \\ x_2(\tau) \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{1+2(0)}+3(0)-1}{2} \\ \frac{1+\sqrt{1+2(0)}}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} =: x^*.$$

It is easy to verify that this point x^* is indeed the solution of $(\text{NLP})_{\geq}$.

Table 5.1 shows the values of $x_1(\tau)$ and $x_2(\tau)$ for a sequence of barrier parameters $\tau = \tau_k$. The initial parameter is related to be $\tau_0 := 1$, and consecutive parameters are decreased by a factor of 10. Observe that $x(\tau) = (x_1(\tau), x_2(\tau))^T$ exhibits a linear rate of convergence to the optimal solution.

□

τ	$x_1(\tau)$	$x_2(\tau)$
10^0	1.8660254	1.3660254
10^{-1}	0.1977226	1.0477226
10^{-2}	0.0199752	1.0049752
10^{-3}	0.0019998	1.0004998
10^{-4}	0.0002000	1.0000500
10^{-5}	0.0000200	1.0000050
10^{-6}	0.0000020	1.0000005

Table 5.1: Barrier function minimizers.

The previous examples illustrates a number of features that typically occur in a barrier method:

(i) $\big(x(\tau_k)\big)_{k\in\mathbb{N}_0}$ converges to a optimal solution x^*. (It is possible to prove convergence for barrier methods under mild conditions.)

(ii) $\tau \mapsto x(\tau)$ $(\tau > 0)$ is a differentiable function, called the *barrier trajectory*. If the logarithmic or inverse barrier method are used, it exists, provided x^* is a regular point of the constraints that satisfies the second-order sufficiency conditions. As well as the strict complementarity conditions. The existence of a trajectory can be used to develop algorithms such as path-following algorithms for LP (similar to Chapter 3). It can also be used to develop techniques that accelerate the convergence of a barrier method.

(iii) Under a regularity condition again, for both the logarithmic and the inverse barrier functions, we find estimates $\mu(\tau)$ of the Lagrange multiplier μ^* at the optimal point x^*. We explain this subsequently.

To (iii): We discuss it for the case of the logarithmic barrier function. Consider a point $x = x(\tau)$ which is a minimizer of the logarithmic barrier function for a specific parameter τ. Setting the gradient of the barrier function (with respect to x) to zero, we obtain

$$\nabla f(x) - \tau \sum_{j=1}^{s} \frac{1}{g_j(x)} \nabla g_j(x) = 0$$

if and only if

$$\nabla f(x) - \sum_{j=1}^{s} \underbrace{\frac{\tau}{g_j(x)}}_{=:\mu_j(\tau)} \nabla g_j(x) = 0$$

(at $x = x(\tau)$). We therefore have a feasible point $x(\tau)$ and a vector $\mu(\tau)$ satisfying the first-order necessary optimality conditions, except τ (instead of 0) on the right-hand side of (cf. $(\mathrm{NOC})^{\tau}_{sf}$ in Section 3.2.):

$$(\mathrm{NOC})^{\tau}_{sf} \qquad \left\{ \begin{array}{c} \nabla f(x(\tau)) - \sum_{j=1}^{s} \mu_j(\tau) \nabla g_j(x(\tau)) = 0 \\ \left. \begin{array}{rcl} \mu_j(\tau) g_j(x(\tau)) & = & \tau \\ \mu_j(\tau) & \begin{array}{c} \geq \\ (>) \end{array} & 0 \end{array} \right\} \forall j \in \{1, 2, \ldots, s\} \\ \\ \text{and (strict) feasibility.} \end{array} \right.$$

As $\tau \to 0^+$, $\mu(\tau)$ can be viewed as an estimate of μ^*, the Lagrange multiplier at x^*. (Indeed, if x^* is a regular point, then: $x(\tau) \to x^*$ $(\tau \to 0) \implies \mu(\tau) \to \mu^*$ $(\tau \to 0)$.)

Example 5.2. Consider the nonlinear problem

$$(\mathrm{NLP})_{\geq} \qquad \left\{ \begin{array}{lrcl} \text{minimize} & f(x) := x_1^2 + x_2^2 & & \\ \text{subject to} & x_1 - 1 & \geq & 0, \\ & x_2 + 1 & \geq & 0, \end{array} \right.$$

the solution being $x^* = (1, 0)^T$. The inequality $j = 1$ is active at x^*, and the corresponding Lagrange multiplier is $\mu_1^* = 2$. Suppose the problem is solved via a logarithmic barrier method. Then, this method solves

$$(\mathrm{NLP})^{k}_{uncon} \qquad \underset{x}{\text{minimize}} \;\; \beta(x, \tau) = x_1^2 + x_2^2 - \tau \ln(x_1 - 1) - \tau \ln(x_2 + 1)$$

for a decreasing sequence of barrier parameters $\tau = \tau_k$ which converge to 0. Now, $(\mathrm{NOC})^{\tau}_{sf}$ gives

$$2x_1 - \frac{\tau}{x_1 - 1} = 0,$$

$$2x_2 - \frac{\tau}{x_2 + 1} = 0,$$

yielding

$$x_1(\tau) = \frac{1 + \sqrt{1 + 2\tau}}{2}, \qquad x_2(\tau) = \frac{-1 - \sqrt{1 + 2\tau}}{2}.$$

The corresponding Lagrange multiplier estimates are:

$$\mu_1(\tau) = \frac{\tau}{x_1(\tau) - 1} = \sqrt{1 + 2\tau} + 1,$$

$$\mu_2(\tau) = \frac{\tau}{x_2(\tau) + 1} = \sqrt{1 + 2\tau} - 1,$$

When $\tau \to 0$, we obtain

$$\lim_{\tau \to 0} \underbrace{\begin{pmatrix} x_1(\tau) \\ x_2(\tau) \end{pmatrix}}_{=x(\tau)} = \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{=:x^*} \quad \text{and} \quad \lim_{\tau \to 0} \underbrace{\begin{pmatrix} \mu_1(\tau) \\ \mu_2(\tau) \end{pmatrix}}_{=\mu(\tau)} = \underbrace{\begin{pmatrix} 2 \\ 0 \end{pmatrix}}_{=:\mu^*}.$$

□

Despite their attractive features, barrier methods also have potential difficulties; we mention the most important one: The unconstrained problems become increasingly difficult to solve as the barrier parameter decreases. The reason is that (with the exception of some special cases) the condition number of the Hessian matrix of the barrier function at its minimum point becomes increasingly large, tending to ∞ as $\tau \to 0$.

Example 5.3. We come back to our Example 5.2. There,

$$\nabla^2_{xx}\beta(x, \tau) = \begin{pmatrix} 2 + \dfrac{\tau}{(x_1 - 1)^2} & 0 \\ 0 & 2 + \dfrac{\tau}{(x_2 + 1)^2} \end{pmatrix}.$$

Suppose now, that $x(\tau)$ is a minimizer of $\beta(\cdot, \tau)$ for some $\tau > 0$. Recall from Example 5.2 that

$$\mu_1(\tau) = \frac{\tau}{x_1(\tau) - 1} \quad \text{and} \quad \mu_2(\tau) = \frac{\tau}{x_2(\tau) + 1}.$$

When $\tau \approx 0$, then, $\mu_1(\tau) \approx 0$. Therefore,

$$\nabla^2_{xx}\beta(x(\tau), \tau) = \begin{pmatrix} 2 + \dfrac{\mu_1^2(\tau)}{\tau} & 0 \\ 0 & 2 + \dfrac{\mu_2^2(\tau)}{\tau} \end{pmatrix} \approx \begin{pmatrix} 2 + \dfrac{4}{\tau} & 0 \\ 0 & 2 \end{pmatrix}.$$

The condition number of this matrix (i.e., the product of both its matrix norm and the one of its inverse; exercise) is approximately equal to

$$\frac{2+\frac{4}{\tau}}{2} = 1 + \frac{2}{\tau} = O(\frac{1}{\tau}),$$

hence, the matrix is ill conditioned (highly sensitive against smallest perturbations). Although the calculations were performed for a point on the barrier trajectory, the same will hold at all points in a neighbourhood of the solution.

The contours of the barrier function in this example are shown in Figure 5.2 for $\tau = 1$ and $\tau = 0.1$. We see that for the smaller τ, the contours (level sets, i.e., the "finger print" of $\beta(\cdot, \tau)$) are almost parallel to the line $x_1 = 1$. More precisely, the contours are almost parallel to the null-space of the gradient of the active constraint at the solution. This is characteristic of barrier functions.

□

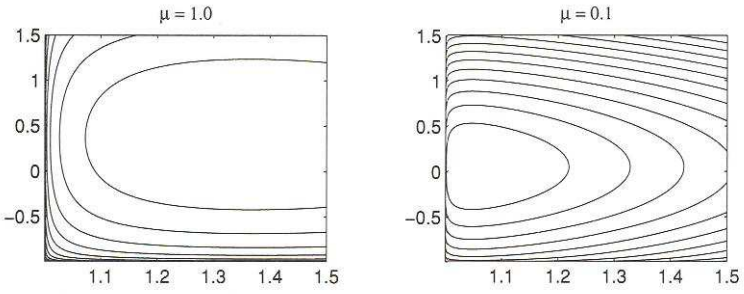


Figure 5.2: Contours of the logarithmic barrier function.

The ill conditioning of the Hessian matrix of $\beta(\cdot, \tau)$ has several ramnifications. It rules out the use of an unconstrained method whose convergence rate depends on the condition number of the Hessian matrix at the solution. Therefore, Newton-type methods are usually the method of choice. The solution to the Newton equations is also sensitive to the ill condition of the Hessian matrix, however. The numerical errors can result in a poor search direction.

In the early 1970s, the ill conditioning of the barrier functions led to their abandonment. Interest in barrier methods was renewed in 1984, with the announcement of Karmarkar's method for LP and the discovery shortly thereafter that this method is just a special case of a barrier method (cf. our discussions above, and Chapter 3)!

Recently, special attention has been given to the development of specialized linear algebra techniques that compute a numerically stable, approximate solution to the Newton equations for a barrier function. We discussed them in Section 3.2 (path-following methods).

Penalty Methods:
In contrast to barrier methods, penalty methods solve a sequence of unconstrained optimization problem whose solution is usually infeasible to the original constrained problem. A penalty for violation of the constraints is incurred, however. As this penalty is increased, the iterates are forced towards the feasible region. An advantage of penalty methods is that they do not require the iterates to be strictly feasible. Thus, unlike barrier methods, they are suitable for problems with equality constraints.

Consider first the equality-constrained problem

$$(\text{NLP})_{=} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0, \end{cases}$$

where f and $h := (h_1, h_2, \ldots, h_m)^T$ are assumed to be of class C^2. The penalty for constraint violation will be a continuous function ψ with the following property:

$$< * > \qquad \begin{array}{l} \psi(x) = 0, \text{ if } x \in M \text{ (feasibility)}, \\ \psi(x) > 0, \text{ otherwise.} \end{array}$$

The best-known such penalty is the quadratic-loss function

$$(a) \qquad \psi(x) = \frac{1}{2}\sum_{i=1}^{m} h_i^2(x) = \frac{1}{2}h^T(x)h(x).$$

Also possible is a penalty of the form

$$(b) \qquad \psi(x) = \frac{1}{\gamma}\sum_{i=1}^{m} |\, h_i^{\gamma}(x)\,|\,, \quad \text{where } \gamma \geq 1.$$

The wight of the penalty is controlled by a *penalty parameter* ρ. As ρ increases, $\rho\psi$ approaches the "ideal penalty" σ: $\rho\psi(x) \to \sigma(x)$ $(\rho \to \infty)$. By adding $\rho\psi$ to f we obtain the penalty function

$$\pi(x, \rho) := f(x) + \rho\psi(x).$$

The penalty method consists in solving a sequence of unconstrained problems of the form

$$(\text{NLP})^k_{uncon} \qquad \underset{x}{\text{minimize}}\, \pi(x, \rho_k)$$

for an increasing sequence $\big(\rho_k\big)_{k\in\mathbb{N}_0}$ of $\rho_k > 0$ tending to infinity:

$$\rho_k \leq \rho_{k+1} \quad \forall k \in \mathbb{N}_0,$$

$$\rho_k \to \infty \quad (k \to \infty).$$

In general, the minimizers of the penalty function violate the constraints $h(x) = 0$. The growing penalty gradually forces these minimizers towards the feasible region.

Penalty methods share many of the advantages of the barrier methods:

(i) Under mild conditions, it is possible to guarantee convergence.

(ii) Under mild conditions, the sequence of penalty function minimizers defines a continuous trajectory.

(iii) In the latter case, it is possible to get estimates of the Lagrange multipliers at the solution.

For example, consider the quadratic-loss penalty function

$$\pi(x, \rho) = f(x) + \frac{1}{2}\rho \sum_{i=1}^{m} h_i^2(x).$$

Its minimizer $x(\rho)$ satisfies

$$\begin{aligned} \nabla_x \pi(x(\rho), \rho) &= \nabla f(x(\rho)) + \rho \sum_{i=1}^{m} \nabla h_i(x(\rho)) h_i(x(\rho)) \\ &= 0, \end{aligned}$$

that is,

$$\nabla f(x(\rho)) - \sum_{i=1}^{m} \underbrace{\Big(- \rho h_i(x(\rho))\Big)}_{=:\lambda_i(\rho)} \nabla h_i(x(\rho)) = 0.$$

If $x(\rho)$ converges to x^* being a solution and regular with respect to the constraints, then, $\lambda(\rho)$ converges to the Lagrange multiplier λ^* associated with x^*:

$$x(\rho) \to x^* \ (\rho \to \infty) \implies \lambda(\rho) \to \lambda^* \ (\rho \to \infty).$$

Penalty functions suffer from the same problems of the ill conditioning as do barrier functions! As ρ increases, the condition number of the Hessian matrix of $\pi(x(\rho), \rho)$ increases, tending to ∞ as $\rho \to \infty$. Therefore the unconstrained minimization can become increasingly difficult to solve.

It is possible to apply penalty methods to problems with inequality constraints. Consider in class C^2, e.g.,

$$(\text{NLP})_{\geq} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & g(x) \geq 0, \end{cases}$$

where $g := (g_1, g_2, \ldots, g_s)^T$. Any continuous function which satisfies $< * >$ can serve as a penalty. Thus, for example, the quadratic-loss penalty in this case is

$$\psi(x) = \frac{1}{2} \sum_{j=1}^{s} \big(\min \{g_j(x), 0\} \big)^2.$$

This function has continuous first derivatives

$$\nabla \psi(x) = \sum_{j=1}^{s} \big(\min \{g_j(x), 0\} \big) \nabla g_j(x),$$

but its second derivatives can be discontinuous at points where some constraint g_j is satisfied exactly. The same observation holds for other simple forms of the penalty function. Thus, we cannot safely use Newton's method to minimize the function. For this reason, straightforward penalty methods have not been widely used for solving general inequality-constrained problems.

Convergence of Penalization Methods:
Let us focus on the convergence of barrier methods when applied to the inequality-constrained problem

$$(\text{NLP})_{\geq} \qquad \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & g(x) \geq 0. \end{cases}$$

We denote the feasible set by M again, and write

$$M_{sf} := \left\{ x \in \mathbb{R}^n \;\middle|\; g_j(x) > 0 \ (j \in \{1, 2, \ldots, s\}) \right\};$$

generically, under some regularity condition: $M_{sf} = M^0$.

In preparing our convergence theorem, we make the following assumptions:

(A1) f and g are continuous.

(A2) $\forall \alpha \in \mathbb{R}: \ \left\{ x \in \mathbb{R}^n \;\middle|\; x \in M, \ f(x) \leq \alpha \right\}$

(A3) $M_{sf} \neq \emptyset$.

(A4) $M = \overline{M_{sf}}$ (topological closure).

Assumptions (A1-2) imply that f has a minimum value on the set M (why?). Assumption (A3) is necessary to define the barrier subproblems. Assumption (A4) is necessary to avoid situations where the minimum point is isolated, and does not have neighbouring interior points.

For example, consider $f(x) := x$ to be minimized subject to the constraints $x^2 - 1 \geq 0$ and $x + 1 \geq 0$. Then

$$M = [1, \infty) \cup \{-1\},$$

where the isolated point $x^* = -1$ is the minimizer, but because it is isolated it is not possible to approach it from the interior of the feasible region, and a barrier method could not converge to this solution. □

The barrier function will be of the form

$$\beta(x, \tau) = f(x) + \tau\varphi(x),$$

where φ can be any function that is continuous on M^0, and that satisfies

$$\varphi(x) \to \infty \quad \text{as} \quad g_j(x) \to 0^+.$$

We will state that under mild conditions, the sequence of barrier minimizers has a convergent subsequence, and the limit of any such convergent subsequence is a solution to the problem. Although in practice convergence of the entire sequence of minimizers is observed, from a theoretical point of view it is not always possible to guarantee convergence of the entire sequence, but only convergence of some subsequence:

Example 5.4. Consider the one-variable problem

$$(\text{NLP})_{\geq} \qquad \begin{cases} \text{minimize} & -x^2 \\ \text{subject to} & 1 - x^2 \geq 0. \end{cases}$$

The logarithmic barrier function is $\beta(x, \tau) = -x^2 - \tau \ln(1 - x^2)$. It has a single minimizer $x^* = 0$ if $\tau = 1$, and two minimizers $x^* = \mp\sqrt{1 - \tau}$ if $\tau < 1$. (The point $x^* = 0$ is a local minimizer if $\tau < 1$) Suppose that $\tau_0, \tau_1, \tau_2, \ldots$ is a sequence of decreasing barrier parameters < 1. Then, a possible sequence of minimizers of $\beta(\cdot, \tau_k)$ is

$$x_k = (-1)^k \sqrt{1 - \tau_k}.$$

This sequence oscillates between neighbourhoods of -1 and $+1$, and, hence, nonconvergent! However, the subsequences $\left(x_{2\nu}\right)_{\nu \in \mathbb{N}_0}$ and $\left(x_{2\nu+1}\right)_{\nu \in \mathbb{N}_0}$ both converge to solutions of $(\text{NLP})_{\geq}$!

□

Theorem 5.1. *Suppose that our given problem* $(\text{NLP})_{\geq}$ *satisfies Assumptions* (A1-4)*, and that a sequence of unconstrained minimization problems*

$$(\text{NLP})^k_{uncon} \qquad \underset{x}{\text{minimize}} \;\; \beta(x, \tau_k) = f(x) + \tau_k \varphi(x)$$

is solved for $0 < \tau_{k+1} < \tau_k \;\forall k \in \mathbb{N}_0$, $\lim\limits_{k \to \infty} \tau_k = 0$. *Furthermore, suppose that the functions* $\beta(\cdot, \tau_k)$ *have a minimum in* $M_{sf} \;\forall k \in \mathbb{N}_0$. *Let this global minimizers be called* x_k.

Then,

(a) $f(x_{k+1}) \leq f(x_k) \quad \forall k \in \mathbb{N}_0$ *(descent);*

(b) $\varphi(x_{k+1}) \geq \varphi(x_k) \quad \forall k \in \mathbb{N}_0$;

(c) $\big(x_k\big)_{k\in\mathbb{N}_0}$ *has a convergent subsequence;*

(d) if $\big(x_{\kappa_\nu}\big)_{\nu\in\mathbb{N}_0}$ *is any convergent subsequence of unconstrained minimizers of* $\beta(\cdot,\tau_{\kappa_\nu})$*, then, its limit point is a global solution* x^* *of* $(\mathrm{NLP})_{\geq}$.

Proof. See Nash, Sofer (1996). □

A key assumption in this theorem is that the barrier functions have a minimizer. We now state a condition which guarantees this:

Lemma 5.1. *Assume that* M *is compact. Then, for any fixed positive value* τ*, there exists a point* $x(\tau) \in M_{sf}$ *which minimizes* $\beta(\cdot,\tau)$.

Proof. See Nash, Sofer (1996), or exercise. □

Another case where the barrier minimizers always exist, is when the problem is convex, even when M is unbounded.

Convergence results for penalty methods can be developed in a similar manner.

Bibliography

[1] G. Golub and C.V. Loan, Matrix Computations. The Johns Hopkins University Press, Baltimore, 1989.

[2] N.I.M. Gould and S. Lyffer, An introduction to algorithms for nonlinear optimization, RAL-TR-2002-031, Computational Science and Engineering Department, Atlas Centre, Rutherford Appleton Laboratory, Oxfordshire, 2003.

[3] H. Hamacher and K. Klamroth, Lineare und Netzwerkoptimierung. Linear and Network Optimization. Ein bilinguales Lehrbuch, Viewegs Verlagsgesellschaft, 2000.

[4] H.Th. Jongen, K. Meer and E. Triesch, Optimization Theory, Kluwer, forthcoming.

[5] S. G. Nash and A. Sofer, Linear and Nonlinear Programming, McGraw-Hill, 1996.

[6] J. Nocedal and S. J. Wright. Numerical Optimization, Springer Series in Operations Research. Springer, 1999.

[7] P. Spellucci, Numerische Verfahren der nichtlinearen Optimierung, ISNM Lehrbuch, Birkhäuser Verlag, 1993.