Universität Bremen*

Optimization of Economic Processes

— Lecture Notes —

Jürgen Pannek

Dynamics in Logistics
Fachbereich 04: Produktionstechnik
Universität Bremen

*EXZELLENT.

Foreword

This script originates from a correspondent lecture held during the summer term 2015 at the University of Bremen. The lecture itself is split into a theoretical and an application part. The theoretical part deals with

- Penalty- and Multiplier-Methods
- SQP and Interior Point Methods
- Integer Optimization and Heuristics

and the application part contains

- Production and Inventory
- Maintenance and Replacement

At the end of the lecture, students should understand the concepts of different kinds of optimization methods and be able to apply these methods to different applications.

Parts of the scripts are based on script of Prof. Gerdts [5] and the books [2, 9], which will be used without further notice. Additional useful information may be found in [3].

Contents

Chapter 1

Introduction

Optimization problem arise in many areas such as econometrics, engineering and natural sciences. In this lecture, we will discuss optimization problems, which are subject to constraints, respective solution methods, and a number of economic applications.

1.1 Problem Setting Optimization

Within the standard setting of this lecture, we suppose functions

$$\begin{aligned} F &: \mathbb{R}^{n_z} \longrightarrow \mathbb{R}, \\ G &: \mathbb{R}^{n_z} \longrightarrow \mathbb{R}^{n_G}, \\ H &: \mathbb{R}^{n_z} \longrightarrow \mathbb{R}^{n_H} \end{aligned}$$

to be given where $\mathbb{R}$ denotes the set of real numbers. We refer to the function F as the *cost function*. The functions G and H are called the *inequality and equality constraints*. These functions shall be sufficiently often continuously differentiable. Within this lecture, we will use the notation for derivatives, which is common in nonlinear optimization. For a continuously differentiable function $g = (g_1, \ldots, g_p) : \mathbb{R}^{n_z} \to \mathbb{R}^p$ we denote the *Jacobian matrix* by

$$\nabla_z g(z) = \begin{pmatrix} \frac{\partial g_1}{\partial z_1} & \cdots & \frac{\partial g_p}{\partial z_1} \\ \vdots & & \vdots \\ \frac{\partial g_1}{\partial z_n} & \cdots & \frac{\partial g_p}{\partial z_n} \end{pmatrix}$$

which we abbreviate to ∇g if there is no ambiguity. For a twice continuously differentiable function $g : \mathbb{R}^{n_z} \to \mathbb{R}$ we write the so called *Hessian* as

$$\nabla^2_{zz} g(z) = \begin{pmatrix} \frac{\partial^2 g}{\partial z_1 z_1} & \cdots & \frac{\partial^2 g}{\partial z_1 z_{n_z}} \\ \vdots & & \vdots \\ \frac{\partial^2 g}{\partial z_{n_z} z_1} & \cdots & \frac{\partial^2 g}{\partial z_{n_z} z_{n_z}} \end{pmatrix}$$

which we abbreviate to $\nabla^2 g$ if there is no danger of confusion.

The argument of the functions F, G, H is called the *optimization variable* and will be denoted by $z \in \mathbb{R}^{n_z}$. Last, we will use the sets $\mathcal{I} = \{1, \ldots, n_G\}$ and $\mathcal{E} = \{1, \ldots, n_H\}$, which we refer to as the *set of inequality and equality constraints*.

Then, we define the standard nonlinear optimization problem (NLP) as follows:

Definition 1.1 (Nonlinear Optimization Problem)
We call the problem

$$\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{with respect to } z \in \mathbb{R}^{n_z} \\ &\text{subject to } G_i(z) \leq 0 \text{ for all } i \in \mathcal{I} \text{ and } H_i(z) = 0 \text{ for all } i \in \mathcal{E} \end{aligned} \tag{NLP}$$

with maps $F : \mathbb{R}^{n_z} \to \mathbb{R}$, $G : \mathbb{R}^{n_z} \to \mathbb{R}^{n_G}$, and $H : \mathbb{R}^{n_z} \to \mathbb{R}^{n_H}$ a *nonlinear optimization problem in standard form.*

The constraints induce the following feasible set:

Definition 1.2 (Feasible Set)
For a problem (NLP) the set

$$\mathcal{F} = \{z \mid G_i(z) \leq 0,\ i \in \mathcal{I};\ H_i(z) = 0,\ i \in \mathcal{E}\} \tag{1.1}$$

is called the *feasible set* and the elements $z \in \mathcal{F}$ are called *feasible points.*

Note that the set $\mathcal{F}$ from Definition 1.2 can only be shown to be closed if the functions G and H are continuous.

The cost function F now allows us to introduce so called *local minimizers*, i.e. points for which the value of the cost function is lower than for surrounding points. These point — at best with the lowest value possible — will be the target points for any of the algorithms we discuss later. Since a minimizer for the problem (NLP) has to be an element of $\mathcal{F}$ by definition, this property need to be included in the definition of a local minimizer in the context of constrained optimization problems:

Definition 1.3 (Local Minimizer)
A point $z^\star \in \mathbb{R}^{n_z}$ is a *local minimizer* of the problem (NLP) if there exists a neighborhood $\mathcal{N}$ of $z^\star$ such that $F(z^\star) \leq F(z)$ holds for all $z \in \mathcal{N} \cap \mathcal{F}$.

Our aim now is to construct numerical methods to compute such a local minimizer $z^\star$ of a problem (NLP). In high school, the problem at hand was (at least) twice continuously differentiable and without constraints. In that case, taking the first derivative and computing its zeros reveals candidates for optimality. Inserting these candidates into the second derivative, local minima, maxima and inflection points can be identified.

The mathematical background of the necessary and sufficient conditions given in respective theorems is Taylor's Theorem, cf. Appendix A.1. Here, we need to include the constraint functions. To find our target $z^\star$, we will require a so called *search direction*. This can be done by arbitrarily picking new candidates and trying to identify areas within which the values of the cost function are particularly low. Or, if the functions F, G, H exhibit differentiability properties, then linear approximations can be used. Before coming to the search direction of the optimization method, we have to know which directions will give us a feasible solution. To this end, also the constraints are linearized

$$G(z+d) \approx G(z) + \nabla G(z)^\top d \quad \text{and} \quad H(z+d) \approx H(z) + \nabla H(z)^\top d.$$

Note that such an approximation makes sense only if the geometry of the feasible set $\mathcal{F}$ is — at least locally — reflected properly when G and H are replaced by approximations. To this end so called *constraint qualifications* are considered in the literature.

The linearized functions allow to introduce the *tangent cone* $T_{\mathcal{F}}(z)$ to the feasible set $\mathcal{F}$.

Definition 1.4 (Tangent Cone)
A vector $v \in \mathbb{R}^{n_z}$ is called *tangent vector* to $\mathcal{F}$ at a point $z \in \mathcal{F}$ if there exists a sequence of feasible points $(z_k)_{k\in\mathbb{N}}$ with $z_k \to z$, $z_k \in \mathcal{F}$ and a sequence of positive scalars $(t_k)_{k\in\mathbb{N}}$ with $t_k \to 0$ such that

$$\lim_{k\to\infty} \frac{z_k - z}{t_k} = v \tag{1.2}$$

holds. The set of all tangent vectors to $\mathcal{F}$ at z is called the *tangent cone* and is denoted by $T_{\mathcal{F}}(z)$.

The tangent cone $T_{\mathcal{F}}$ depends on the geometry of $\mathcal{F}$ only. At a given feasible point $z \in \mathcal{F}$, the set $T_{\mathcal{F}}(z)$ can be seen as a local approximation of all feasible directions, i.e. all vectors $d \in \mathbb{R}^{n_z}$ for which $z + \alpha d \in \mathcal{F}$ holds for all sufficiently small $\alpha > 0$. The definition of $T_{\mathcal{F}}$ implies that each feasible direction is contained in $T_{\mathcal{F}}(z)$. Conversely, for each element $v \in T_{\mathcal{F}}(z)$ and each $\epsilon > 0$ there exists a feasible direction d with $\|d - v\| < \epsilon$.

We directly observe that all equality constraints H_i restrict the set of feasible directions. Yet this is not necessarily the case for all inequality constraints: If $G_i(z) > 0$ holds, then we can utilize continuity of G_i to get $G_i(z+\alpha d) > 0$ for all $d \in \mathbb{R}^{n_z}$ provided $\alpha > 0$ is sufficiently small. If, however, $G_i(z) = 0$ holds, then an arbitrarily small change of z in the "wrong" direction may lead to $G_i(z + \alpha d) < 0$. Hence, the latter inequality constraints also restrict the set of feasible directions. This gives rise to the so called *active set* and the respective *active constraints*:

Definition 1.5 (Active Set)
The *active set* $\mathcal{A}(z)$ at any feasible point z consists of the equality constraint indices from $\mathcal{E}$ together with the indices of the inequality constraints $i \in \mathcal{I}$ where $G_i(z) = 0$ holds, that is $\mathcal{A}(z) := \mathcal{E} \cup \{i \in \mathcal{I} \mid G_i(z) = 0\}$.

Definition 1.6 (Active Constraints)
Consider the active set $\mathcal{A}(z)$ of a feasible point $z \in \mathcal{F}$. Then we call

$$A(z) := \begin{pmatrix} (G_i)_{i\in\mathcal{A}(z)\cap\mathcal{I}} \\ (H_i)_{i\in\mathcal{E}} \end{pmatrix} \tag{1.3}$$

the set or vector of active constraints and $n_A = \sharp\mathcal{A}(z)$ the number or dimension of active constraints at z. Moreover, we denote the corresponding Lagrange multiplier vector by λ^A.

1.1.1 Necessary Conditions for Optimality

The center of many numerical algorithms for computing the optimum of a nonlinear optimization problem (NLP) are the so called *Karush–Kuhn–Tucker* (KKT) conditions. To state these

conditions, we introduce the *Lagrangian* $L : \mathbb{R}^{n_z} \times \mathbb{R}^{1+n_G+n_H} \to \mathbb{R}$. For its definition, we require the Lagrange multipliers $\lambda_0 \in \mathbb{R}$, $\lambda \in \mathbb{R}^{n_H}$ and $\mu \in \mathbb{R}^{n_G}$ and define the Lagrangian as a modification of the cost function F by

$$L(z, \lambda_0, \lambda, \mu) := \lambda_0 F(z) + \lambda^\top G(z) + \mu^\top H(z). \tag{1.4}$$

Note that the additional terms $\lambda^\top G(z) + \mu^\top H(z)$ penalize violations of the constraints.

Before coming to the general case of a nonlinear optimization problem, let us consider the more simple convex case, i.e.

minimize $F(z)$
with respect to $z \in \mathbb{R}^{n_z}$
subject to $G_i(z) \leq 0$ for all $i \in \mathcal{I}$ and $H_i(z) = Az - b = 0$ for all $i \in \mathcal{E}$

where we assume the feasible set $\mathcal{F}$ to be nonempty. From standard calculus we know that the set $\mathcal{F}$ is convex. Moreover, since F is convex, then also the set of global minima is convex, and local minima are also global ones. Additionally, we can formulate necessary and sufficient conditions rather simple:

Theorem 1.7 (Fritz John Conditions – Necessary Conditions for the Convex Case)
Let $z^\star$ be optimal for a convex optimization problem. Then there exist non trivial Lagrange multipliers $(\lambda_0, \lambda, \mu) \in \mathbb{R}^{1+n_G+n_H}$ such that the following conditions hold:

- *Sign condition:*

$$\lambda_0 \geq 0, \qquad \lambda_i \geq 0, \qquad i = 1, \ldots, n_G \tag{1.5}$$

- *Minimality of the Lagrangian:*

$$L(z^\star, \lambda_0, \lambda, \mu) \leq L(z, \lambda_0, \lambda, \mu) \qquad \forall z \in \mathcal{F} \tag{1.6}$$

- *Complementarity condition:*

$$\lambda_i G_i = 0, \qquad i = 1, \ldots, n_G \tag{1.7}$$

- *Feasibility:*

$$z^\star \in \mathcal{F} \tag{1.8}$$

Theorem 1.8 (Sufficient Conditions for the Convex Case)
Suppose $z^\star \in \mathcal{F}$ is given. If conditions (1.5) – (1.8) *hold with $\lambda_0 = 1$, then $z^\star$ is optimal.*

Since we know how to deal with unconstrained optimization problems, we like to reduce the constrained one to an unconstrained one and apply known methods to it. To this end, we make use of the active set $\mathcal{A}$, which represent all constraints that are satisfied with equality. We solve these restrictions for some components of z, and optimize over the components that are left. We illustrate this via an example.

Example 1.9
In order to produce tins, two different materials are used for the lids and the shell, which costs p_1 and p_2 units per square unit respectively. The aim is to produce the tins for a given volume $V > 0$ at cheapest cost.
Formulation of the (NLP):

1. *The lids are circular with radius $r > 0$ and area $r^2\pi$. Hence the costs are $2p_1r^2\pi$.*
2. *The area of the shell measures $2r\pi h$, where $h > 0$ is the height of the tin. The costs are given by $2p_2r\pi h$.*
3. *The volume of the tin is given by $r^2\pi h$.*

Hence, we have

$$\begin{aligned} &\text{minimize} \quad F(z) = 2p_1r^2\pi + 2p_2r\pi h \\ &\text{with respect to } z = (r, h) \in \mathbb{R}^2 \\ &\text{subject to } H(z) = r^2\pi h - V = 0. \end{aligned}$$

Solution of the equality restriction:
If $r \neq 0$, then the constraint $H(r, h) = r^2\pi h - V = 0$ can be reformulated as

$$h(r) = \frac{V}{r^2\pi}. \tag{1.9}$$

The case $r = 0$ can be ruled out since the condition $V > 0$ cannot be met. For $h(r)$ we then have

$$H(r, h(r)) = 0.$$

Inserting (1.9) into F, we obtain the equivalent optimization problem

$$\begin{aligned} &\text{minimize} \quad F(r, h(r)) = 2p_1r^2\pi + \frac{2Vp_2}{r} \\ &\text{with respect to } z = r \in \mathbb{R}. \end{aligned}$$

Computing the optimum:
We apply the known first order necessary conditions to $F(r, h(r))$ to obtain a candidate. Differentiating $F(r, h(r))$ gives us

$$\frac{dF}{dr}(r, h(r)) = \frac{\partial F}{\partial r}(r, h(r)) + \frac{\partial F}{\partial h}(r, h(r)) \cdot \frac{\partial h}{\partial r}(r). \tag{1.10}$$

To evaluate this expression, we differentiate $H(r, h(r)) = 0$ with respect to r using the chain rule which gives us

$$0 = \frac{\partial H}{\partial r}(r, h(r)) + \frac{\partial H}{\partial h}(r, h(r)) \cdot \frac{\partial h}{\partial r}(r).$$

Since $\frac{\partial H}{\partial h}(r, h(r)) = r^2\pi \neq 0$ for $r \neq 0$, we can solve the latter for $\frac{\partial h}{\partial r}(r)$ and obtain

$$\frac{\partial h}{\partial r}(r) = -\left(\frac{\partial H}{\partial h}(r, h(r))\right)^{-1} \frac{\partial H}{\partial r}(r, h(r)) = -\frac{1}{r^2\pi} 2r\pi h(r) = -\frac{2V}{r^3\pi}. \tag{1.11}$$

Inserting (1.11) *into* (1.10) *and setting* (1.10) *equal to zero gives us*

$$0 = \frac{dF}{dr}(r, h(r)) = 4r\pi p_1 - \frac{2Vp_2}{r^2} \tag{1.12}$$

and reveals the positive solution

$$r = \sqrt[3]{\frac{Vp_2}{2\pi p_1}}, \quad h(r) = \frac{V}{r^2\pi}.$$

Since the cost function F is convex, this solution represents the minimum.

Alternative solution:
We define the Lagrange multiplier

$$\lambda := -\frac{\partial F}{\partial h}(r, h(r)) \left(\frac{\partial H}{\partial h}(r, h(r))\right)^{-1} \tag{1.13}$$

and insert (1.13) *into* (1.10) *which gives us*

$$0 = \frac{\partial F}{\partial r}(r, h(r)) + \lambda \frac{\partial H}{\partial r}(r, h(r)).$$

Moreover, (1.13) *is equivalent to*

$$0 = \frac{\partial F}{\partial h}(r, h(r)) + \lambda \frac{\partial H}{\partial h}(r, h(r))$$

Using the Lagrangian, these conditions can be written as

$$\begin{aligned} 0 &= \frac{\partial L}{\partial r}(r, h, \lambda) \\ 0 &= \frac{\partial L}{\partial h}(r, h, \lambda) \\ 0 &= H(r, h) \end{aligned}$$

representing the so called Lagrangian multiplier rule. These conditions form a nonlinear equation system, and its solution corresponds to the one from the first approach.

To state the KKT conditions in the convex case, one typically introduces the *Slater condition*

$$\exists z \in \mathcal{F} : \; G(z) < 0. \tag{1.14}$$

Note that if no Slater point exists, then only the Fritz–John conditions hold. Since the cost function F is not present in these conditions due to $\lambda_0 = 0$, the Fritz–John conditions can also be seen as degenerate KKT conditions.

The approach we followed in Example 1.9 utilized one of the fundamental theorems of calculus, and is not limited to the convex case but applies for the general nonlinear case as well.

Theorem 1.10 (Implicit Function Theorem)
Suppose $H : \mathbb{R}^{n_z - n_H} \times \mathbb{R}^{n_H}$ *to be continuously differentiable and* $(\eta^\star, \theta^\star) \in \mathbb{R}^{(n_z - n_H) + n_H}$ *to satisfy* $H(\eta^\star, \theta^\star) = 0$. *If the* $p \times p$ *matrix* $\frac{\partial H}{\partial \theta}(\eta^\star, \theta^\star)$ *is invertible, then there exist neighborhoods* $B_\varepsilon(\eta^\star)$ *and* $B_\delta(\theta^\star)$ *with radii* $\varepsilon, \delta > 0$ *and a mapping* $\theta : B_\varepsilon(\eta^\star) \to \mathbb{R}^{n_H}$ *with* $\theta(\eta^\star) = \theta^\star$ *and*

$$H(\eta, \theta(\eta)) = 0 \qquad \forall (\eta, \theta(\eta)) \in B_\varepsilon(\eta^\star) \times B_\delta(\theta^\star).$$

Moreover, $\theta(\cdot)$ *is continuously differentiable in* $B_\varepsilon(\eta^\star)$ *and the Jacobian of* $\theta(\cdot)$ *is given by*

$$\frac{d\theta}{d\eta}(\eta) = -\left(\frac{\partial H}{\partial \theta}(\eta, \theta)\right)^{-1} \cdot \frac{\partial H}{\partial \eta}(\eta, \theta) \qquad \forall (\eta, \theta) \in B_\varepsilon(\eta^\star) \times B_\delta(\theta^\star).$$

Tracking along the footsteps of Example 1.9, we can define the Lagrange multiplier λ via

$$\lambda^\top := -\frac{\partial F}{\partial \theta}(\eta^\star, \theta^\star) \cdot \left(\frac{\partial H}{\partial \theta}(\eta^\star, \theta^\star)\right)$$

and the Lagrangian via

$$L(z, \lambda) := F(z) + \lambda^\top H(z) \qquad \text{with } z = (\eta, \theta)^\top,$$

which allows us to apply first order necessary condition for an unconstrained problem revealing

Theorem 1.11 (Lagrange multiplier rule)
Consider $F : \mathbb{R}^{n_z} \to \mathbb{R}$ *and* $H : \mathbb{R}^{n_z} \to \mathbb{R}^{n_H}$ *to be continuously differentiable. Suppose* $z^\star$ *to be a minimizer of* F *with* $H(z^\star) = 0$ *and* $rang(dH(z^\star)/dz) = n_H$. *Then there exists a Lagrange multiplier* $\lambda \in \mathbb{R}^{n_H}$ *satisfying*

$$0 = \nabla_z L(z^\star, \lambda) = \nabla F(z^\star) + \frac{dH}{dz}(z^\star)^\top \lambda.$$

Now, the nonlinear equation system

$$\nabla_z L(z, \lambda) = 0, \qquad H(z) = 0$$

can be solved for z and λ using, e.g., Newton's method, which leads to the so called *Lagrange–Newton Method*.

For the more general nonlinear case, the *Linear Independent Constraint Qualification* LICQ is used. To define this condition, we utilize the active set, which basically plays this case back to the one with equality constraints only. We first introduce the set of "linearized" feasible directions obtained from the linearizations of G.

Definition 1.12 (Linearized Feasible Directions)
For a feasible point $z \in \mathcal{F}$ and the active set $\mathcal{A}(z)$ we call the set

$$\mathcal{F}(z) = \left\{ v \in \mathbb{R}^{n_z} \;\middle|\; \begin{array}{l} v^\top \nabla G_i(z) \leq 0 \text{ for all } i \in \mathcal{A}(z) \cap \mathcal{I} \text{ and} \\ v^\top \nabla H_i(z) = 0 \;\; \text{for all } i \in \mathcal{E} \end{array} \right\} \tag{1.15}$$

the *set* (or cone) *of linearized feasible directions.*

Since $T_{\mathcal{F}}(z) \subseteq \mathcal{F}(z)$ and we want to show necessary optimality conditons based on linearizations, these sets should coincide. This is the intention of constraint qualifications, i.e., that the geometry of $T_{\mathcal{F}}$ is captured by the linearizations of G_i and H_i. The linear independence constraint qualification is probably the most popular one.

Definition 1.13 (LICQ)
Consider a feasible point z and the active set $\mathcal{A}(z)$. Suppose that F, H and G are continuously differentiable. If the elements of the gradient set $\{\nabla G_i(z) \mid i \in \mathcal{A}(z) \cap \mathcal{I}\} \cup \{\nabla H_i(z) \mid i \in \mathcal{E}\}$ are linearly independent then we say that the *linear independence constraint qualification* (LICQ) holds.

Under this condition we obtain $T_{\mathcal{F}}(z) = \mathcal{F}(z)$, see [3, Lemma 9.2.1].

Similar to the Lagrange multiplier rule from Theorem 1.11, we can now state a first order necessary optimality condition — usually called *KKT (Karush–Kuhn—Tucker) condition* — for the constrained case, which will serve as a guideline to find local minimizers, see [3, Theorem 9.1.1].

Theorem 1.14 (KKT Conditions)
Consider the problem (NLP) with local minimizer $z^\star \in \mathcal{F}$. Moreover suppose the functions F, G and H to be continuously differentiable and the (LICQ) to hold at $z^\star$. Then there exists Lagrange multiplier $\lambda^\star \in \mathbb{R}^{n_G}$, $\mu^\star \in \mathbb{R}^{n_H}$ such that the following conditions hold.

$$\nabla_z L(z^\star, \lambda^\star, \mu^\star) = 0 \tag{1.16}$$
$$G_i(z^\star) \leq 0 \qquad \forall i \in \mathcal{I} \tag{1.17}$$
$$H_i(z^\star) = 0 \qquad \forall i \in \mathcal{E} \tag{1.18}$$
$$\lambda_i^\star \geq 0 \qquad \forall i \in \mathcal{I} \tag{1.19}$$
$$\lambda_i^\star G_i(z^\star) = 0 \qquad \forall i \in \mathcal{I} \tag{1.20}$$
$$\mu_i^\star H_i(z^\star) = 0 \qquad \forall i \in \mathcal{E} \tag{1.21}$$

The identity (1.20) is a so called strict complementarity condition which says that either $\lambda_i^\star = 0$ or $G_i(z^\star) = 0$ must hold. A special case which is important for nonlinear optimization algorithms is the following.

Definition 1.15
Consider the problem (NLP) with local minimizer $z^\star \in \mathcal{F}$ and Lagrange multipliers $\lambda^\star \in \mathbb{R}^{n_G}$, $\mu^\star \in \mathbb{R}^{n_H}$ satisfying (1.16) - (1.21). Then we say that the *strict complementarity condition* holds if $\lambda_i^\star > 0$ for all $i \in \mathcal{I} \cap \mathcal{A}(z^\star)$.

We see that the KKT conditions connect the gradient of the cost function to active constraints. In particular, Theorem 1.14 states that for a given minimizer $z^\star$ moving along an arbitrary vector $v \in \mathcal{F}(z^\star)$ either increases the value of the first order approximation of the cost function, i.e. $v^\top \nabla F(z^\star) > 0$, or keeps its value at the same level in the case $v^\top \nabla F(z^\star) = 0$.

In the second case, it is unknown if the cost function value is increasing or decreasing along v. Here, second order conditions can be used to obtain more information about change of F, see [3, Theorem 9.3.1] for a corresponding proof.

Theorem 1.16 (Second Order Necessary Conditions)
Consider the problem (NLP) with local minimizer $z^\star \in \mathcal{F}$. Suppose the functions F, G and H to be continuously differentiable and the (LICQ) to hold at $z^\star$. Let $\lambda^\star \in \mathbb{R}^{n_G}$, $\mu^\star \in \mathbb{R}^{n_H}$ be Lagrange multipliers satisfying the KKT conditions (1.16)–(1.21). *Then the inequality*

$$v^\top \nabla^2_{zz} L(z^\star, \lambda^\star, \mu^\star) v \geq 0 \tag{1.22}$$

holds for all

$$v \in \mathcal{C}(z^\star, \lambda^\star) := \left\{ v \in \mathcal{F}(z^\star) \;\middle|\; \begin{array}{l} v^\top \nabla G_i(z^\star) = 0 \text{ for all} \\ i \in \mathcal{A}(z^\star) \cap \mathcal{I} \text{ with } \lambda_i^\star > 0 \end{array} \right\}. \tag{1.23}$$

1.1.2 Sufficient Conditions for Optimality

The set $\mathcal{C}$ is also called the *critical cone*. It contains all directions which leave the active inequality constraints with $\lambda_i > 0$ as well as all equality constraints active if one moves a sufficiently small step along these directions. This, however, does not need to hold for those active inequality constraints with $\lambda_i = 0$. In particular, we have the equivalence

$$v \in \mathcal{C}(z^\star, \lambda^\star) \Longleftrightarrow \left\{ \begin{array}{ll} \nabla G_i(z^\star)^\top v = 0, & \text{for all } i \in \mathcal{A}(z^\star) \cap \mathcal{I} \text{ with } \lambda_i^\star > 0, \\ \nabla G_i(z^\star)^\top v \leq 0, & \text{for all } i \in \mathcal{A}(z^\star) \cap \mathcal{I} \text{ with } \lambda_i^\star = 0, \\ \nabla H_i(z^\star)^\top v = 0, & \text{for all } i \in \mathcal{E}. \end{array} \right.$$

Now, we want to get a converse result, i.e. we want to check whether a given feasible point is actually a local minimizer. As it turns out, the only differences between the previous necessary conditions and the sufficient conditions presented next is that the constraint qualification is not required whereas inequality (1.22) needs to be strengthened to a strict inequality, cf. [3, Theorem 9.3.2]:

Theorem 1.17 (Second Order Sufficient Conditions)
Consider a feasible point $z^\star \in \mathcal{F}$ and suppose Lagrange multiplier $\lambda^\star \in \mathbb{R}^{n_G}$, $\mu^\star \in \mathbb{R}^{n_H}$ to exist satisfying (1.16) – (1.21). *If we have*

$$v^\top \nabla^2_{zz} L(z^\star, \lambda^\star, \mu^\star) v > 0 \tag{1.24}$$

for all $v \in \mathcal{C}(z^\star, \lambda^\star)$ with $v \neq 0$, then $z^\star$ is a strict local minimizer of problem (NLP).

1.2 Problem Setting Economic Processes

In contrast to the previous Section 1.1, we want to consider processes in our applications, i.e. dynamic systems and not static problems. Here, we define a process to be driven by a model, which is a a discrete or continuous time control system. And our aim in this section is to play this dynamic problem back to a static one in order to apply methods based on the theory from Section 1.1.

First, we need to introduce a basic definition of a our variables:

Definition 1.18 (Time set)
A *time set* $\mathcal{T}$ is a subgroup of $(\mathbb{R}, +)$.

By setting $\mathcal{T} = \mathbb{Z}$ or $\mathcal{T} = \mathbb{R}$, we can formally switch between discrete and continuous time. Having defined time, we now introduce the states and controls of a system:

Definition 1.19 (State and Control)
We call the set $\mathcal{U}$ the *control set* and the set $\mathcal{X}$ the *state set*. Moreover, the set of all maps from a set $\mathcal{I} \subset \mathcal{T}$ to a set $\mathcal{U}$ is denoted by $U^{\mathcal{I}} = \{u \mid u : \mathcal{I} \to \mathcal{U}\}$ and called the *set of control functions*. The elements $x \in \mathcal{X}$ and $u \in \mathcal{U}$ are called *state* and *control* of a system.

Given time, states and control, we can now define their connection via a dynamic system:

Definition 1.20 (Discrete time Control System)
Consider a function $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. A system of difference equations

$$x_u(k+1, x_0) := f(x_u(k, x_0), u(k)), \qquad k \in \mathbb{N}_0 \tag{1.25}$$

is called a *discrete time control system*. Moreover $x_u(k, x_0) \in \mathcal{X}$ is called *state vector* and $u(k) \in u$ *control vector*.

Existence and uniqueness of a solution of (1.25) is clear by induction. In particular, we obtain a unique solution in positive time direction for a certain maximal existence interval.

In the continuous time setting, a control system is given as follows:

Definition 1.21 (Continuous time Control System)
Consider a function $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. A system of first order ordinary differential equations

$$\dot{x}_u(t) = f(x_u(t, x_0), u(t)), \qquad t \in \mathbb{R} \tag{1.26}$$

is called a *continuous time control system*.

The control system itself only gives us the state change over time. To compute a possible future trajectory, we require additional information on the starting point.

Definition 1.22 (Initial Value Condition)
Consider a point $x_0 \in \mathcal{X}$. Then the equation

$$x(0) = x_0 \in \mathcal{X} \tag{1.27}$$

is called the *initial value condition*.

Note that existence and uniqueness of a trajectory is guaranteed if the system is Lipschitz or if the requirements of Caratheodory's Theorem are met, cf. [6] and [10] respectively. Utilizing existence and uniqueness, we can introduce the notion of a trajectory or solution:

Definition 1.23 (Solution)
We call the unique function $x_u(t, x_0)$ a *solution* for $t \in \mathcal{T}$ if it satisfies the initial value condition (1.27) and the control system equation (1.25) or (1.26).

Similar to the static case, we assign costs to a trajectory of the control system. In principle, this simple fact already removes the dynamics from our problem by simply considering the entire time stream as an optimization variable. This brings us to the notion of a so called *optimal control problem*. The costs are given via the functional

$$J_N(x_0, u) = \sum_{k=0}^{N-1} \ell\left(x_u(k, x_0), u(k)\right) + L(x_u(N, x_0)) \tag{1.28}$$

where $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ and $L : \mathcal{X} \to \mathbb{R}$ are the so called stage and terminal costs. A typical choice of these functions is the quadratic version

$$\ell(x, u) = \|x\|^2 + \lambda\|u\|^2, \qquad L(x) = \|x\|^2.$$

Note that computing a control

$$u^\star = \operatorname*{argmin}_{u \in U^N} J_N(x_0, u) \tag{1.29}$$

may not be tractable if N is very large or even $N = \infty$.

For control systems (1.25) or (1.26), constraints are motivated by boundaries of processes, e.g. that there exists only a finite number of gears in a gearbox or that the capacity of a road is bounded. The most general approach to incorporate constraints in the control system setting is via sets:

Definition 1.24 (Constraints)
For given state and control sets $\mathcal{X}$ and $\mathcal{U}$, we call the subsets

$$\mathbb{X} \subset \mathcal{X} \qquad \text{and} \qquad \mathbb{U} \subset \mathcal{U} \tag{1.30}$$

the *constrained state* and *control sets*.

Based on these constraints, we can now introduce the concept of *feasibility sets*. Since we have to anticipate future events in the state space, feasibility sets require us to change the perspective in time. Hence, a reverse time view is needed. This leads to the following definition:

Definition 1.25 (Feasible Set and Admissible Set)
Consider a control system (1.25) (1.30) and $\mathbb{X}^0 \subset \mathbb{X}$. For any time frame $\mathcal{I} = [0, N] \subset \mathbb{N}_0$ the *feasible set* is defined via

$$\mathbb{X}^N := \{x_0 \mid \exists u : \ x_u(N, x_0) \in \mathbb{X}^0, \ x_u(k, x_0) \in \mathbb{X}, \ u(k) \in \mathbb{U} \ \forall k \in \{0, \dots, N-1\}\}. \tag{1.31}$$

Moreover, the *admissible set* is given by

$$\mathbb{U}^N_{\mathbb{X}^N}(x_0) := \{u \mid x_u(N, x_0) \in \mathbb{X}^0, \ x_u(k, x_0) \in \mathbb{X}, \ u(k) \in \mathbb{U} \ \forall k \in \{0, \dots, N-1\}\}. \tag{1.32}$$

The difference between feasibility and admissibility is the following:

> Admissibility deals with controls, feasibility is about states. In particular, a control is called admissible for a specific state. And a state is called feasible if there exists a control sequence such that future states satisfy the state constraints.

Last, we can combine the cost functional (1.28) with the control system dynamics (1.25), the initial value condition (1.27) and the feasibility condition (1.31) to obtain an dynamic equivalent to our nonlinear optimization problem (NLP):

Definition 1.26 (Optimal Control Problem (OCP))
We call the problem

$$\begin{aligned} &\text{Minimize} \quad J_N(x_0, u) := \sum_{k=0}^{N-1} \ell(x_u(k), u(k)) + L(x_u(N, x_0)) \\ &\text{with respect to} \quad u(\cdot) \in \mathbb{U}_{\mathbb{X}^N}^N(x_0), \quad \text{subject to} \\ &x_u(0, x_0) = x_0 \in \mathbb{X}^0, \quad x_u(k+1, x_0) = f(x_u(k, x_0), u(k)) \end{aligned} \tag{OCP}$$

an *optimal control problem.*

Remark 1.27
If the time period between two time instances is fixed to T, we can obtain the equivalent continuous time optimal control problem by replacing (1.25) *by* (1.26) *and the cost* (1.28) *by*

$$J_N(x_0, u) = \int_{t=0}^{NT} \ell\left(x_u(t, x_0), u(t)\right)\, dt + L(x_u(NT, x_0)).$$

The aim of the remainder of this section we give the foundations of the so called "first discretize then optimize" approach. In a transformation step, we discretize the optimal control problem (OCP) into a nonlinear optimization problem in standard form (NLP).

Remark 1.28
Apart from the "first discretize then optimize" approach there also exists a so called "first optimize then discretized" method. Applying the latter requires in deep knowledge of Pontryagin's minimum principle. The basic idea is to introduce the adjoint differential equation as part of an integrated solution. Yet, this approach is no universal remedy as computing the solution of this approach requires numerical methods similar to optimization methods as well.

1.2.1 Discretization Methods

Even though (OCP) is already a discrete time problem, the process of converting (OCP) into (NLP) is called *discretization*. Here, we will stick with this commonly used term while in a strict sense we only convert one discrete problem into another.

As we will see, the (NLP) problem related to (OCP) can be formulated in different ways. The first variant, called *full discretization*, incorporates the dynamics (1.25) as additional constraints into (NLP). This approach is very straightforward but causes large computing times for solving the problem (NLP) due to its dimensionality.

The second approach is designed to deal with this dimensionality problem. It recursively computes $x_u(k, x_0)$ from the dynamics (1.25) outside of the optimization problem (NLP), thus reducing the number of constraints. However, this so called *recursive discretization* has some drawbacks regarding parallelization, warm start and sensitivity.

1.2.2 Full Discretization

Within the full discretization technique, the trajectory $x_u(k, x_0)$ in (OCP) is given by the dynamics (1.25) or (1.26). Now, each control value $u(k)$, $k \in \{0, \dots, N-1\}$ is an optimization variable in (OCP) and also an optimization variable in (NLP). The idea of the full discretization is now to treat each point on the trajectory $x_u(k, x_0)$ as an additional independent n_x-dimensional optimization variable and define the total optimization variable via

$$z := (x_u(0, x_0)^\top, \dots, x_u(N, x_0)^\top, u(0)^\top, \dots, u(N-1)^\top)^\top. \tag{1.33}$$

To guarantee that the solution of (NLP) also corresponds to a trajectory of (1.25), we add respective equality constraints to (NLP), which read

$$x_u(k+1, x_0) - f(x_u(k, x_0), u(k)) = 0 \qquad \text{for } k \in \{0, \dots, N-1\} \tag{1.34}$$

$$x_u(0, x_0) - x_0 = 0 \tag{1.35}$$

Additionally, we have to reformulate the constraints $u \in \mathbb{U}^N_{\mathbb{X}^N}(x_0)$, which can be written as

$$\begin{aligned} x_u(k, x_0) &\in \mathbb{X} && k \in \{0, \dots, N\} \\ u(k) &\in \mathbb{U} && k \in \{0, \dots, N-1\} \end{aligned} \tag{1.36}$$

Note that the setting is easily extended to the case of time varying constraints.

In the following, we assume $\mathbb{X}$ and $\mathbb{U}$ to be given by a set of functions

$$\begin{aligned} G_i^S &: \mathbb{R}^n_x \times \mathbb{R}^n_u \to \mathbb{R}, \quad i \in \mathcal{I}^S = \{1, \dots, n_G\} \\ H_i^S &: \mathbb{R}^n_x \times \mathbb{R}^n_u \to \mathbb{R}, \quad i \in \mathcal{E}^S = \{1, \dots, n_H\} \end{aligned}$$

via equality and inequality constraints of the form

$$G_i^S(x_u(k, x_0), u(k)) \le 0, \quad i \in \mathcal{I}^S,\ k \in K_i \subseteq \{0, \dots, N\} \tag{1.37}$$

$$H_i^S(x_u(k, x_0), u(k)) = 0, \quad i \in \mathcal{E}^S,\ k \in K_i \subseteq \{0, \dots, N\}. \tag{1.38}$$

where the index sets K_i, $i \in \mathcal{I}^S \cup \mathcal{E}^S$ formalize the possibility that some of these constraints are not required at time instant $k \in \{0, \dots, N\}$. This reveals the following:

Definition 1.29 (Full Discretization)
The nonlinear programming problem in standard form (NLP)

$$\text{Minimize} \quad F(z) := \sum_{k=0}^{N-1} \ell(x_u(k, x_0), u(k)) + L(x_u(N, x_0))$$

with respect to

$$z := (x_u(0, x_0)^\top, \dots, x_u(N, x_0)^\top, u(0)^\top, \dots, u(N-1)^\top)^\top \in \mathbb{R}^{n_z}$$

subject to $G(z) = \left[G_i^S(x_u(k, k), u(k))\right]_{i \in \mathcal{I}^S, k \in K_i} \le 0$

and $H(z) = \begin{bmatrix} \left[H_i^S(x_u(k, x_0), u(k))\right]_{i \in \mathcal{E}^S, k \in K_i} \\ \left[x_u(k+1, x_0) - f(x_u(k, x_0), u(k))\right]_{k \in \{0, \dots, N-1\}} \\ x_u(0, x_0) - x_0 \end{bmatrix} = 0$

is called the full discretization of Problem (OCP).

The advantage of the full discretization is its simplicity. On the backside, the method results in a high dimensional optimization variable $z \in \mathbb{R}^{(N+1)\cdot n_x + N\cdot n_u}$ and a large number of both equality and inequality constraints. Since computing times of solvers for (NLP) depend massively on the size of the problem, this is unwanted.

1.2.3 Recursive Discretization

The methodology of the recursive discretization is inspired by the (hierarchical) divide and conquer principle. Basically, the control system dynamics is decoupled and treated as a sub-problem of the optimization problem. These two layers exchange information regarding the control sequence u and the initial value x_0 from the (NLP) to the simulation, and the state sequences $x_u(\cdot, x_0)$ in the opposite direction.

The optimization variable z reduces to

$$z := (u(0)^\top, \dots, u(N-1)^\top)^\top \tag{1.39}$$

and the constraint functions $H_i^S : \mathbb{R}_x^n \times \mathbb{R}_u^n \to \mathbb{R}$, $i \in \mathcal{E}^S$ are given by (1.37). The inequality constraints $G_i^S : \mathbb{R}_x^n \times \mathbb{R}_u^n \to \mathbb{R}$, $i \in \mathcal{I}^S$ and the cost function F remain unchanged. Hence, the recursively discretized problem takes the following form:

Definition 1.30 (Recursive Discretization)
The nonlinear programming problem in standard form (NLP)

$$\begin{aligned}
&\text{minimize} \quad F(z) := \sum_{k=0}^{N-1} \ell(x_u(k, x_0), u(k)) + L(x_u(N, x_0)) \\
&\text{with respect to } z := (u(0)^\top, \dots, u(N-1)^\top)^\top \in \mathbb{R}^{n_z} \\
&\text{subject to } H(z) = \left[H_i^S(x_u(k, x_0), u(k))\right]_{i \in \mathcal{E}^S, k \in K_i} = 0 \\
&\text{and } G(z) = \left[G_i^S(x_u(k, x_0), u(k))\right]_{i \in \mathcal{I}^S, k \in K_i} \geq 0
\end{aligned}$$

is called the recursive discretization of Problem (OCP).

Analyzing the dimension of the optimization variable and the number of equality constraints, we see that using the recursive discretization the optimization variable consists of $N \cdot n_u$ scalar components and the number of equality constraints is reduced to the number of conditions in (1.37). We can conclude that this discretization is minimal in these regards.

Unfortunately, the method has some drawbacks regarding parallelization, warm start and sensitivity. These shortcomings can to some extend be circumvented by incorporating multiple shooting techniques, which are beyond the scope of this lecture. The basic idea is to find a suitable compromise between the full and the recursive discretization by introducing few breaking points into the recursive discretization.

Part I

Optimization

Chapter 2

Penalty– and Multiplier–Methods

Within this chapter, we discuss the popular Penalty– and Multiplier–Methods, which are based on coupling the constraints to the cost function via weighted penalty terms. The penalty term penalizes inadmissible points. The advantage of the method lies the removal of constraints, which allows for a direct use of algorithms from unconstrained optimization. Here, we do not display any proofs and instead refer to the book [4], which serves as the basis of this chapter. Further details may also be obtained from the book [9].

The concept of Penalty–Methods for the general task

$$\boxed{\begin{aligned}&\text{minimize} \quad F(z)\\ &\text{with respect to } z \in \mathcal{F} \subset \mathbb{R}^{n_z}.\end{aligned}} \tag{PP}$$

works as follows: First, we require a function $r : \mathbb{R}^{n_z} \to \mathbb{R}_0^+$ such that

$$r(z) \begin{cases} = 0, & \text{if } z \in \mathcal{F} \\ > 0, & \text{if } z \notin \mathcal{F}. \end{cases}$$

Then, for a suitably chosen sequence of weighting parameters $(\eta^{[k]})_{k\in\mathbb{N}}$ with $\eta^{[k]} > 0$ we minimize the unconstrained penalty function

$$P(z; \eta_k) := F(z) + \eta^{[k]} r(z). \tag{2.1}$$

For each $\eta^{[k]} > 0$ we obtain a solution $z^{[k]} := z(\eta^{[k]})$ and we need to ask how the weighting parameters $\eta^{[k]}$, $k \in \mathbb{N}$ have to be chosen for the sequence $(z^{[k]})_k$ to converge to a minimum of the original problem (PP).

The function r can be defined in many ways. Differentiable functions are ideal as they allow for the usage of known methods for unconstrained optimization. In case r is continuous but not continuously differentiable, the solution of the penalty problem is more involved.

2.1 Penalty–Methods

Let's start this section with an example:

Example 2.1
Consider the optimization problem

$$\textit{Minimize} \quad F(z_1, z_2) := z_1 + z_2 \quad \textit{subject to} \quad H(z_1, z_2) := z_1^2 - z_2 = 0.$$

Now we want to eliminate the constraint. To achieve the latter, we could solve the constraint function for z_2 and insert it into the cost function as illustrated in the Lagrange approach in the previous Chapter 1. Here, we want to couple a penalty term to the cost function, which penalizes point satisfying $z_1^2 - z_2 \neq 0$. One such function is given by

$$r(z_1, z_2) := (z_1^2 - z_2)^2 = H(z_1, z_2)^2.$$

This function realizes $r(z_1, z_2) = 0$ if and only if $H(z_1, z_2) = 0$. Note that r is differentiable. We could also have used the absolute value $|r(z_1, z_2)|$ instead of the square, but this function is not differentiable.
Instead of F, we now consider the penalty function

$$P(z_1, z_2, \eta) := F(z_1, z_2) + \frac{\eta}{2} r(z_1, z_2) = z_1 + z_2 + \frac{\eta}{2} \left(z_1^2 - z_2\right)^2,$$

where $\eta > 0$ is the weighting parameter.
We can now apply methods for unconstrained optimization, but the question remains on how the weighting parameter η influences the solution, and under which conditions the solutions converge to the solution of the original problem. To this end, we first consider the necessary conditions

$$0 = \nabla_z P(z_1, z_2, \eta) = \begin{pmatrix} 1 + 2\eta z_1 \left(z_1^2 - z_2\right) \\ 1 - \eta \left(z_1^2 - z_2\right) \end{pmatrix}.$$

cf. Theorem A.2. From these conditions, we obtain the stationary points

$$\begin{pmatrix} z_1(\eta) \\ z_2(\eta) \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{4} - \frac{1}{\eta} \end{pmatrix}.$$

To see how these point correlate with the solutions of the original problem, we first compute the stationary points of the Lagrangian

$$L(z_1, z_2, \lambda) = z_1 + z_2 + \lambda \left(z_1^2 - z_2\right),$$

which are given by

$$0 = \nabla_z L(z_1, z_2, \lambda) = \begin{pmatrix} 1 + 2\lambda z_1 \\ 1 - \lambda \end{pmatrix} \qquad \Longleftrightarrow \qquad \begin{pmatrix} z_1(\lambda) \\ z_2(\lambda) \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{4} \end{pmatrix} \text{ with } \lambda = 1.$$

For the latter, we observe that the solutions of the Penalty–Problem (PP) converge to the solution of the constrained problem for $\eta \to \infty$.

For more general results, we consider the equality constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{with respect to } z \in \mathcal{F} = \{z \in \mathbb{R}^{n_z} \mid H_i(z) = 0,\ i = 1, \ldots, n_H\}. \end{aligned} \tag{PPE}$$

where all functions $z : \mathbb{R}^{n_z} \to \mathbb{R}$ and $H_i : \mathbb{R}^{n_z} \to \mathbb{R}$, $i = 1, \ldots, n_H$ are continuous. The idea of the penalty method is to approximate the solution $z^\star$ of the original problem (PPE) iteratively by a series of unconstrained auxiliary problems. The latter problems consist in minimizing the

penalty function

$$P(z,\eta) = F(z) + \frac{\eta}{2}\sum_{i=1}^{n_H} (H_i(z))^2 \tag{2.2}$$

for suitable values of $\eta > 0$. By attaching the constraints to the cost, leaving the feasible set $\mathcal{F}$ is penalized. The constant η represents a weighting factor, which can be used to adapt the intensity of the penalization. The Penalty method is given by the following algorithm.

Algorithm 2.2 (Penalty Method)
Suppose a pair of initial values $(z^{[0]}, \eta^{[0]})$ to be given and set $k := 0$.
While $H(z^{[k]}) \not\approx 0$ do

1. Compute solution $z^{[k]}$ of

$$\text{minimize} \quad P(z,\eta^{[k]}) = F(z) + \frac{\eta^{[k]}}{2}\sum_{i=1}^{n_H} (H_i(z))^2 \quad \text{over} \quad z \in \mathbb{R}^{n_z}$$

2. Determine $\eta^{[k+1]} > \eta^{[k]}$ and set $k := k+1$

Since (PPE) is not differentiable in general, we require methods from unconstrained non differentiable optimization to solve the minimization of (2.2) in Step 1 of Algorithm 2.2. Here, the question arises whether such a method actually converges to the solution of problem (PPE). The answer to that is given in the following theorem:

Theorem 2.3 (Convergence of the Penalty Method)
Suppose F and H_i, $i = 1, \ldots, n_H$ to be continuous functions and $\left(\eta^{[k]}\right)_k$ to be strictly monotone increasing with $\eta^{[k]} \to \infty$. Moreover, consider the feasible set $\mathcal{F}$ to be nonempty and $\left(z^{[k]}\right)_k$ is a sequence generated by Algorithm 2.2. Then the following holds:

1. *The sequence of penalty function values $\left(P(z^{[k]}, \eta^{[k]})\right)_k$ is monotone increasing.*
2. *The sequence of violations of constraints $\left(\|H(z^{[k]})\|\right)_k$ is monotone decreasing.*
3. *The sequence of cost function values $\left(F(z^{[k]})\right)_k$ is monotone increasing.*
4. *We have $\lim_{k\to\infty} H(z^{[k]}) = 0$.*
5. *Each limit point of the sequence $\left(z^{[k]}\right)_k$ is a solution of (PPE).*

Within problem (PPE) we considered equality constraints only. However, we only required these function to be continuous, which also applies for the modification

$$\max\{0, G_i(z)\} = 0, \quad i = 1, \ldots, n_G$$

of the inequality constraints G_i, $i = 1, \ldots, n_G$ and allow us to simply extend the penalty function (2.2) to

$$P(z,\eta) = F(z) + \frac{\eta}{2}\sum_{i=1}^{n_H} (H_i(z))^2 + \frac{\eta}{2}\sum_{i=1}^{n_G} (\max\{0, G_i(z)\})^2 .$$

The main disadvantage of the Penalty method is the fact that the weighting factor η must tend to ∞ to obtain convergence of the method. This leads to ill-conditioned problems in Step 1 of Algorithm 2.2.

Note that so far we didn't state how the weighting factor $\eta^{[k+1]}$ shall be determined in Step 2 of Algorithm 2.2. To derive the latter, we analyze how we can construct a sequence $\left(\eta^{[k]}\right)_k$ along $\left(z^{[k]}\right)_k$ such that both sequences converge to a KKT point $(z^\star, \lambda^\star)$ of the original problem (PPE). To this end, we require continuous differentiability of the functions F and H_i, $i = 1, \ldots, n_H$. A KKT point $(z^\star, \lambda^\star)$ satisfies

$$0 = \nabla F(z^\star) + \lambda_i^\star \sum_{i=1}^{n_H} \nabla H_i(z^\star).$$

Since $z^{[k]}$ is a minimum of P, we have that

$$0 = \nabla_z P(z^{[k]}, \eta^{[k]}) = \nabla F(z^{[k]}) + \eta^{[k]} \sum_{i=1}^{n_H} H_i(z^{[k]}) \nabla H_i(z^{[k]}).$$

Comparing these expressions, it seems promising to choose

$$\lambda_i^{[k]} = \eta^{[k]} H_i(z^{[k]}) \tag{2.3}$$

as an approximation of the Lagrange multipliers $\lambda_i^\star$. For this choice, the following result holds true:

Theorem 2.4 (Convergence of Adjoints)
Consider F and H_i, $i = 1, \ldots, n_H$ to be continuous functions and $\left(z^{[k]}\right)_k$ to be a sequence generated by Algorithm 2.2 with $z^{[k]} \to z^\star$ for $k \to \infty$. Moreover, the gradients $\nabla H_i(z^\star)$, $i = 1, \ldots, n_H$ are linear independent and the sequence $\left(\lambda^{[k]}\right)_k$ is given by (2.3). Then, the following holds:

1. *The sequence $\left(\lambda^{[k]}\right)_k$ converges to a vector $\lambda^\star$.*
2. *$(z^\star, \lambda^\star)$ is a KKT point of the original problem (PPE).*

At the same time, (2.3) gives rise to the determination of the weighting factor via

$$\eta^{[k+1]} = \eta^{[k]} \sum_{i=1}^{n_H} H_i(z^{[k]})$$

2.2 Multiplier–Penalty–Methods

Multiplier–Penalty methods are similar to Penalty methods, but utilize exact and differentiable penalty functions — the so called *Lagrange function*. Again, we consider the equality constrained problem

$$\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{with respect to } z \in \mathcal{F} = \{z \in \mathbb{R}^{n_z} \mid H_i(z) = 0,\ i = 1, \ldots, n_H\}. \end{aligned} \tag{PPE}$$

where all functions $z : \mathbb{R}^{n_z} \to \mathbb{R}$ and $H_i : \mathbb{R}^{n_z} \to \mathbb{R}$, $i = 1, \ldots, n_H$ are twice continuously

differentiable. Suppose $z^\star$ is a local minimum of (PPE). Then, for $\eta > 0$ we have that $z^\star$ is also a local minimum for

$$\text{minimize} \quad F(z) + \frac{\eta}{2}\|H(z)\|^2 \quad \text{over } z \in \mathbb{R}^{n_z} \quad \text{such that} \quad H(z) = 0.$$

The Lagrangian of this problem is given by

$$L_a(z, \lambda, \eta) := F(z) + \frac{\lambda}{2}\|H(z)\|^2 + \eta^\top H(z)$$

and is called *extended* or *augmented Lagrangian* or *Multiplier–Penalty–Function*. It can be shown, that the weighting factor η within L_a is not required to tend to infinity in order to obtain a local minimum of the original problem (PPE).

Lemma 2.5
Suppose $(z^\star, \lambda^\star)$ is a KKT point of (PPE). Moreover, the second order sufficient conditions from Theorem 1.17 hold. Then there exists a finite $\eta_0 > 0$ such that $z^\star$ is a strict local minimum of $L_a(\cdot, \lambda^\star, \eta)$ for all $\eta \geq \eta_0$.

As a conclusion from Lemma 2.5, we can solve the original problem (PPE) indirectly via

$$\begin{aligned} &\text{minimize} \quad L_a(z, \lambda^\star, \eta) \\ &\text{with respect to } z \in \mathbb{R}^{n_z}. \end{aligned} \tag{PPA}$$

The penalty parameter η is not required to tend to ∞ as it is the case for the Penalty method from Algorithm 2.2. Additionally, L_a is differentiable, which allows us to apply known methods from unconstrained optimization.

Unfortunately, the optimal Lagrange multiplier $\lambda^\star$ is unknown. To approximate the latter, we suppose η to be sufficiently large and $z^{[k]}$ to be a stationary point of

$$\text{minimize} \quad L_a(z, \lambda^{[k]}, \eta) \quad \text{over } z \in \mathbb{R}^{n_z}.$$

Necessary condition now read

$$0 = \nabla_z L_a(z^{[k+1]}, \lambda^{[k]}, \eta) = \nabla F(z^{[k+1]}) + \sum_{i=1}^{n_H} \left(\lambda_i^{[k]} + \eta H_i(z^{[k+1]})\right) \nabla H_i(z^{[k+1]}).$$

Moreover, for a KKT point $(z^\star, \lambda^\star)$ of (PPE), condition

$$0 = \nabla_z L(z^\star, \lambda^\star) = \nabla F(z^\star) + \sum_{i=1}^{n_H} \lambda_i^\star \nabla H_i(z^\star)$$

must necessarily hold. Comparing the last two expressions, we obtain the updating technique

$$\lambda^{[k+1]} := \lambda^{[k]} + \eta H(z^{[k+1]}]) \tag{2.4}$$

which gives rise to the following algorithm.

Algorithm 2.6 (Multiplier–Penalty Method)
Suppose a pair of initial values $(z^{[0]}, \eta^{[0]})$, a weight $\eta^{[0]} > 0$ and a $\sigma \in (0, 1)$ to be given and set $k := 0$.
While $(z^{[k]}, \eta^{[k]})$ is not a KKT point of (PPE) do

1. Compute solution $z^{[k+1]}$ of (PPA)
$$\text{minimize} \quad L_a(z, \lambda^{[k]}, \eta^{[k]}) \quad \text{over} \quad z \in \mathbb{R}^{n_z}$$
2. Set $\lambda^{[k+1]}$ according to (2.4)
$$\lambda^{[k+1]} := \lambda^{[k]} + \eta^{[k]} H(z^{[k+1]})$$
3. If $\|H(z^{[k+1]})\| \geq \sigma \|H(z^{[k]})\|$, then set $\eta^{[k+1]} = 10\eta^{[k]}$, otherwise set $\eta^{[k+1]} = \eta^{[k]}$
4. Set $k := k + 1$

We can extend our setting (PPE) to include inequality constraints

$$\text{minimize} \quad F(z) \quad \text{over } z \in \mathbb{R}^{n_z} \quad \text{such that} \quad G(z) \leq 0,\ H(z) = 0.$$

To this end, we introduce slack variables $s = (s_1, \ldots, s_{n_G}) \in \mathbb{R}^{n_G}$ and obtain

$$\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{over } (z, s) \in \mathbb{R}^{n_z + n_G} \\ &\text{such that } G_i(z) + s_i^2 = 0,\ i = 1, \ldots, n_G \\ &\qquad\qquad H_i(z) = 0,\ i = 1, \ldots, n_H \end{aligned}$$

The augmented Lagrangian of this problem is given by

$$L_a(z, s, \lambda, \mu, \eta) = F(z) + \frac{\eta}{2}\|H(z)\|^2 + \mu^\top H(z) + \sum_{i=1}^{n_G} \left(\lambda (G_i(z) + s_i^2) + \frac{\eta}{2}(G_i(z) + s_i^2)^2 \right).$$

For a given z, we can explicitly solve this minimization with respect to s and obtain

$$s_i = \left(\max\left(0, -\left(\frac{\lambda_i}{\eta} + G_i(z)\right)\right)\right)^{1/2}, \quad i = 1, \ldots, n_G.$$

Inserting the latter in the augmented Lagrangian we see

$$\begin{aligned} L_a(z, \lambda, \mu, \eta) &= F(z) + \mu^\top H(z) + \frac{\eta}{2}\|H(z)\|^2 + \frac{1}{2\eta}\sum_{i=1}^{G} \left((\max\{0, \lambda_i + \eta G_i(z)\})^2 - \lambda_i^2 \right) \\ &= F(z) + \sum_{i=1}^{n_H} \left(\mu_i H_i(z) + \frac{\eta}{2} H_i(z)^2 \right) \\ &\quad + \sum_{i=1}^{n_G} \begin{cases} \lambda_i G_i(z) + \frac{\eta}{2} G_i(z)^2, & \text{if } \lambda_i + \eta G_i(z) \geq 0 \\ -\frac{\lambda_i^3}{2\eta}, & \text{else} \end{cases} \end{aligned}$$

Note that this function is only continuously differentiable once. For the multipliers, we obtain the following updating formulas:

$$\begin{aligned} \mu^{[k+1]} &:= \mu^{[k]} + \eta H(z^{[k+1]}]), \\ \lambda^{[k+1]} &:= \max\left(0, \lambda^{[k]} + \eta G(z^{[k+1]})\right), \end{aligned}$$

Chapter 3

SQP and Interior Point Methods

Within this chapter, we discuss two methods, which can be termed state–of–the–art in nonlinear optimization at present. Since the research field for these methods — the so called Sequential Quadratic Programming approach (SQP) and the Interior Point Method (IP) — are quite active, we focus on the basics of these methods only. For deeper insights, we refer to the books [4, 9], which also serve as sources for proofs of theorems stated in this chapter.

3.1 Sequential Quadratic Programming

To motivate the *sequential quadratic programming approach* (SQP), we discuss the so called *Lagrange–Newton method*. This method is suitable to solve optimization problem, which are subject to equality constraints

$$\boxed{\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{with respect to } z \in \mathcal{F} = \{z \in \mathbb{R}^{n_z} \mid H_i(z) = 0,\ i = 1, \ldots, n_H\} \end{aligned}} \tag{PPE}$$

where the functions $F : \mathbb{R}^{n_z} \to \mathbb{R}$ and $H : \mathbb{R}^{n_z} \to \mathbb{R}^{n_H}$ are twice continuously differentiable and $L(z, \lambda) = F(z) + \lambda^\top H(z)$ is the Lagrange function. The Lagrange–Newton method applies Newton's method to the KKT conditions

$$\nabla_z L(z, \lambda) = 0 \qquad \text{and} \qquad H(z) = 0$$

and reads as follows:

Algorithm 3.1 (Lagrange–Newton Method)
Suppose $z^{[0]} \in \mathbb{R}^{n_z}$, $\lambda^{[0]} \in \mathbb{R}^{n_H}$ and $\varepsilon > 0$ to be given and set $k = 0$.
While $\max\{\|\nabla_z L(z^{[k]}, \lambda^{[k]})\|, \|H(z^{[k]})\|\} > \varepsilon$ do

1. Solve the linear equation system

$$\begin{pmatrix} \nabla^2_{zz} L(z^{[k]}, \lambda^{[k]}) & \nabla_z H(z^{[k]})^\top \\ \nabla_z H(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} d \\ v \end{pmatrix} = - \begin{pmatrix} \nabla_z L(z^{[k]}, \lambda^{[k]}) \\ H(z^{[k]}) \end{pmatrix} \tag{3.1}$$

2. Set

$$z^{[k+1]} := z^{[k]} + d \qquad \text{and} \qquad \lambda^{[k+1]} := \lambda^{[k]} + v \tag{3.2}$$

3. Set $k := k + 1$

Alternatively, the Lagrange–Newton method can be introduced using a quadratic approximation of the cost function, which is also referred to as the direct approach. Utilizing the KKT conditions is known as the indirect approach. Note that both ideas result in the same algorithm.

3.1.1 Quadratic Approximation

The alternative approach deals with the approximation

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} d^\top \nabla_{zz} L(z^{[k]}, \lambda^{[k]}) d + \nabla_z F(z^{[k]})^\top d \\ &\text{with respect to } d \in \mathbb{R}^{n_z} \\ &\text{subject to } H(z^{[k]}) + \nabla_z H(z^{[k]}) d = 0. \end{aligned} \tag{QPE}$$

The Lagrangian for this quadratic problem is given by

$$L_{(QP)}(d, \mu) := \frac{1}{2} d^\top \nabla_{zz} L(z^{[k]}, \lambda^{[k]}) d + \nabla_z F(z^{[k]})^\top d + \mu^\top \left(H(z^{[k]}) + \nabla_z H(z^{[k]}) d \right).$$

Now, applying the KKT conditions reveals the linear equation system

$$\begin{aligned} \nabla_{zz} L(z^{[k]}, \lambda^{[k]}) d + \nabla_z F(z^{[k]}) + \nabla_z H(z^{[k]})^\top \mu &= 0 \\ H(z^{[k]}) + \nabla_z H(z^{[k]}) d &= 0 \end{aligned}$$

or equivalently

$$\begin{pmatrix} \nabla_{zz} L(z^{[k]}, \lambda^{[k]}) & \nabla_z H(z^{[k]})^\top \\ \nabla_z H(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} d \\ \mu \end{pmatrix} = - \begin{pmatrix} -\nabla_z F(z^{[k]}) \\ H(z^{[k]}) \end{pmatrix}. \tag{3.3}$$

Subtracting $\nabla_z H(z^{[k]})^\top \mu^{[k]}$ from both sides of the first equation in (3.3) now reveals

$$\begin{pmatrix} \nabla_{zz} L(z^{[k]}, \lambda^{[k]}) & \nabla_z H(z^{[k]})^\top \\ \nabla_z H(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} d \\ \mu - \lambda^{[k]} \end{pmatrix} = - \begin{pmatrix} -\nabla_z L(z^{[k]}, \lambda^{[k]}) \\ H(z^{[k]}) \end{pmatrix}, \tag{3.4}$$

which is equivalent to (3.1) with $v = \mu - \lambda^{[k]}$. Hence, the new iterates can be evaluated via

$$z^{[k+1]} := z^{[k]} + d \qquad \text{and} \qquad \lambda^{[k+1]} := \mu. \tag{3.5}$$

This gives rise to the following conclusion:

Conclusion 3.2
For equality constraint problems (PPE), the Lagrange–Newton method is equivalent to the sequential quadratic optimization method displayed above if the multiplier μ in the quadratic auxiliary problem is chosen as the new approximation of the multiplier λ of problem (PPE).

3.1.2 SQP Algorithm

Utilizing this conclusion, we can apply the approximation idea to our standard optimization problem (NLP) from Definition 1.1, which gives us

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} d^\top \nabla_{zz} L(z^{[k]}, \lambda^{[k]}) d + \nabla_z F(z^{[k]})^\top d \\ &\text{with respect to } d \in \mathbb{R}^{n_z} \\ &\text{subject to } G(z^{[k]}) + \nabla_z G(z^{[k]}) d \leq 0, \\ &\qquad\qquad H(z^{[k]}) + \nabla_z H(z^{[k]}) d = 0. \end{aligned} \tag{QP}$$

To play this problem back to an equality constrained one (QPE), we introduce the constraint function $C : \mathbb{R}^{n_z} \to \mathbb{R}^{n_H + n_G}$, which combines the constraints G_i and H_i into one function

$$C : z \mapsto \begin{bmatrix} (G_i(z))_{i \in \mathcal{I}} \\ (H_i(z))_{i \in \mathcal{E}} \end{bmatrix}.$$

Now, we can define the Lagrangian via

$$L(z, \lambda) := F(z) + \lambda^\top C(z). \tag{3.6}$$

Then, we introduce a so called *working set* $\mathcal{W}_k$ of the current operating point z_k. This working set contains all indexes of constraints which are currently active, that is all equality constraints $i \in \mathcal{E}$ and all inequality constraints $i \in \mathcal{I}$ satisfying equality. Note that this is similar to the active constraints introduced in Definition 1.5. Yet, in order to update the working set, the entire combination of constraints is more useful. For the working set $\mathcal{W}_k$, the constraints are linearized and the cost functional is approximated using a second order Taylor approximation of the Lagrangian, which reveals

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} {d^{[k]}}^\top \nabla^2_{zz} L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}}) d^{[k]} + \nabla_z F(z^{[k]})^\top d^{[k]} \\ &\text{with respect to } d^{[k]} \in \mathbb{R}^{n_z} \\ &\text{subject to } C_i(z^{[k]}) + \nabla_z C_i(z^{[k]})^\top d^{[k]} = 0 \text{ for all } i \in \mathcal{W}^{[k]} \end{aligned} \tag{SQP}$$

We can solve this problem by computing the solution of the linear equation

$$\begin{pmatrix} \nabla^2_{zz} L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}}) & \nabla_z C_{\mathcal{W}^{[k]}}(z^{[k]})^\top \\ \nabla_z C_{\mathcal{W}^{[k]}}(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} d^{[k]} \\ \lambda^{[k]}_{\mathcal{W}^{[k]}} \end{pmatrix} = - \begin{pmatrix} \nabla_z F(z^{[k]}) \\ C_{\mathcal{W}^{[k]}}(z^{[k]}) \end{pmatrix} \tag{3.7}$$

The next iterate is then given by

$$z_{k+1} := z_k + d_k \qquad \text{and} \qquad \lambda^{[k+1]} := \lambda^{[k]}.$$

At each iterate, the working set is updated and a new search direction step is computed until the first order optimality conditions are satisfied sufficiently well.

Hence, we obtain the following algorithm:

Algorithm 3.3 (Local SQP Method)
Suppose $\mathcal{W}^{[0]}$, $z^{[0]} \in \mathbb{R}^{n_z}$, $\lambda^{[0]}_{\mathcal{W}^{[0]}} \in \mathbb{R}^{n_G + n_H}$ and $\varepsilon > 0$ to be given and set $k := 0$.
While $\max\{\|\nabla_z L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}})\|, \|\lambda^{[k]}_{\mathcal{W}^{[k]}} C_{\mathcal{W}^{[k]}}(z^{[k]})\|\} > \varepsilon$ do

1. Solve the linear equation system

$$\begin{pmatrix} \nabla^2_{zz} L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}}) & \nabla_z C_{\mathcal{W}^{[k]}}(z^{[k]})^\top \\ \nabla_z C_{\mathcal{W}^{[k]}}(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} d^{[k]} \\ \lambda^{[k]}_{\mathcal{W}^{[k]}} \end{pmatrix} = - \begin{pmatrix} \nabla_z F(z^{[k]}) \\ C_{\mathcal{W}^{[k]}}(z^{[k]}) \end{pmatrix}$$

and obtain $d^{[k]}$, $\lambda^{[k]}$ and $\mathcal{W}^{[k+1]}$

2. Set

$$z^{[k+1]} := z^{[k]} + d^{[k]} \qquad \text{and} \qquad \lambda^{[k+1]} := \lambda^{[k]}$$

3. Set $k := k + 1$

Within Algorithm 3.3, a priori knowledge of the index set $\mathcal{W} = \mathcal{A}(z^\star)$ is not required. However, the iterates $z^{[k]}$ are typically not feasible. Regarding convergence, the following result holds:

Theorem 3.4 (Convergence of the local SQP method)
Suppose the following holds:

- *$z^\star$ is a local minimum of our standard problem (NLP) and $\lambda^\star$, $\mu^\star$ denote the respective Lagrange multipliers.*
- *The functions F, G_i, $i \in \mathcal{I}$, H_i, $i \in \mathcal{E}$ are twice continuously differentiable and the second order derivatives are Lipschitz.*
- *LICQ holds.*
- *The strict complementarity condition $\lambda_i^\star - G_i(z^\star) > 0$ holds for all $i \in \mathcal{A}(z^\star)$.*
- *The second order sufficient condition*

$$d^\top \nabla_{zz}^2 L(z^\star, \lambda^\star, \mu^\star) d > 0$$

holds for all $d \neq 0$ satisfying

$$\nabla_z G_i(z^\star)^\top d = 0,\ i \in \mathcal{A}(z^\star) \qquad \text{and} \qquad \nabla_z H_i(z^\star)^\top d = 0,\ i \in \mathcal{E}.$$

Then there exist neighborhoods $\mathcal{U}$ of $(z^\star, \lambda^\star, \mu^\star)$ and $\mathcal{V}$ of $(0, \lambda^\star, \mu^\star)$ such that for arbitrary initial values

$$(z^{[0]}, \lambda^{[0]}, \mu^{[0]}) \in \mathcal{U}$$

all problems (QP) possess a unique local solution

$$(d^{[k]}, \lambda^{[k+1]}, \mu^{[k+1]}) \in \mathcal{V}.$$

Moreover, the solution converges quadratically to $(z^\star, \lambda^\star, \mu^\star)$.

As the result shows, the SQP method converges for all initial values in a neighborhood of the local minimum. Yet, this neighborhood can be very small. Therefore it is necessary to globalize the SQP method so that it converges for arbitrary initial values. As in the unconstrained case, this can be done by introducing a step size $\alpha^{[k]} > 0$, and defining the new iterate via

$$z^{[k+1]} := z^{[k]} + \alpha^{[k]} d^{[k]}.$$

To obtain the step size $\alpha^{[k]}$, a one–dimensional line search is executed. However, it is not clear whether $z^{[k+1]}$ is „better“ than $z^{[k]}$. The reason for this lies in the construction of the iterates:

The iterates shall improve the costs and the constraint violations, which may be contradicting goals.

3.1.3 Globalization of SQP

To avert this dilemma, a *merit function* is introduced, which at simplest is a combination of the cost function and the constrained violation, cf. the idea of the penalty function in Chapter 2. Based on this merit function, an improvement can be measured. The general class of merit functions is defined by

$$P(z,\eta) := F(z) + \eta r(z) \tag{3.8}$$

where $\eta > 0$ is a weighting parameter and $r : \mathbb{R}^{n_z} \to \mathbb{R}_0^+$ is a continuous function satisfying

$$r(z) \begin{cases} = 0, & \text{if } z \in \mathcal{F} \\ > 0, & \text{if } z \notin \mathcal{F}. \end{cases}$$

Of particular interest are the so called exact merit functions. For these functions, the local minima of the restricted original problem (NLP) are also local minima of the unconstrained merit function, and the weighting factor η can be chosen to be finite.

Definition 3.5 (Exact Merit Function)
The merit function $P(z,\eta)$ from (3.8) is called exact in a local minimum $z^\star$ of (NLP), if there exists a finite parameter $\eta^\star > 0$ such that $z^\star$ is a local minimum of $P(\cdot,\eta)$ for all $\eta \geq \eta^\star$.

It would be nice if a differentiable exact merit function was available. Unfortunately, one can show that $P(z,\eta)$ from (3.8) is not differentiable in $z^\star$ if $P(z,\eta)$ is exact and $\nabla_z F(z^\star) \neq 0$, which is the usual case in constrained optimization. Still, one can show the following:

Theorem 3.6 (Exact Merit Function)
Suppose $z^\star \in \mathcal{F}$ is an isolated local minimum of (NLP) satisfying the Linear Independent Constraint Qualification LICQ from Definition 1.13. Then the merit function ℓ_q with

$$\ell_q(z,\eta) := F(z) + \eta \left(\sum_{i=1}^{n_G} (\max\{0, G(z)\})^q + \sum_{i=1}^{n_H} |H_i(z)|^q \right)^{1/q}, \qquad 1 \leq q < \infty$$

$$\ell_\infty(z,\eta) := F(z) + \eta \max\{0, G_1(z), \ldots, G_{n_G}(z), |H_1(z)|, \ldots, |H_{n_H}(z)|\}$$

is exact for $1 \leq q \leq \infty$.

Here, we restrict ourselves to ℓ_1–merit functions. We can assume that for sufficiently large $\eta > 0$ the constrained problem (NLP) can be replaced by the unconstrained problem

minimize $\quad \ell_1(z,\eta)$
with respect to $z \in \mathbb{R}^{n_z}$.

This idea can be used in the SQP method to compute the step size $\alpha^{[k]}$ via the one dimensional line search regarding the function

$$\varphi(\alpha^{[k]}) := \ell_1(z^{[k]} + \alpha^{[k]} d^{[k]}, \eta).$$

Although ℓ_1 is not differentiable, it is still directionally differentiable, i.e. the limit value of

$$\nabla_z \ell_1(z^{[k]}, \eta) := \lim_{\alpha \to 0} \frac{\ell_1(z^{[k]} + \alpha^{[k]} d^{[k]}, \eta) - \ell_1(z^{[k]}, \eta)}{\alpha^{[k]}}$$

exists for all $z \in \mathbb{R}^{n_z}$ and all $d \in \mathbb{R}^{n_z}$. Moreover, one can show that a KKT point $(d^{[k]}, \lambda^{[k]}, \mu^{[k]})$ with $d^{[k]} \neq 0$ of (QP) satisfies the estimate

$$\nabla_z \ell_1(z^{[k]}, \eta) \leq -{d^{[k]}}^\top \nabla^2_{zz} L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}}) d^{[k]} < 0$$

if the Hessian is symmetric positive definite and if the weighting parameter is chosen such that

$$\eta \geq \max\{\lambda_1^{[k+1]}, \ldots, \lambda_{n_G}^{[k+1]}, |\mu_1^{[k+1]}|, \ldots, |\mu_{n_H}^{[k+1]}|\}.$$

Combined, we obtain the following algorithm:

Algorithm 3.7 (Global SQP Method)
Suppose $\mathcal{W}^{[0]}$, $z^{[0]} \in \mathbb{R}^{n_z}$, $\lambda^{[0]}_{\mathcal{W}^{[0]}} \in \mathbb{R}^{n_G + n_H}$ and $\varepsilon > 0$, $\sigma \in (0,1)$ to be given and set $k := 0$.
While $\max\{\|\nabla_z L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}})\|, \|\lambda^{[k]}_{\mathcal{W}^{[k]}} C_{\mathcal{W}^{[k]}}(z^{[k]})\|\} > \varepsilon$ do

1. Solve the linear equation system

$$\begin{pmatrix} \nabla^2_{zz} L(z^{[k]}, \lambda^{[k]}_{\mathcal{W}^{[k]}}) & \nabla_z C_{\mathcal{W}^{[k]}}(z^{[k]})^\top \\ \nabla_z C_{\mathcal{W}^{[k]}}(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} d^{[k]} \\ \lambda^{[k]}_{\mathcal{W}^{[k]}} \end{pmatrix} = - \begin{pmatrix} \nabla_z F(z^{[k]}) \\ C_{\mathcal{W}^{[k]}}(z^{[k]}) \end{pmatrix}$$

   and obtain $d^{[k]}$, $\lambda^{[k]}$ and $\mathcal{W}^{[k+1]}$

2. Choose $\eta^{[k+1]} \geq \max\{\eta^{[k]}, \lambda_1^{[k+1]}, \ldots, \lambda_{n_G}^{[k+1]}, |\mu_1^{[k+1]}|, \ldots, |\mu_{n_H}^{[k+1]}|\}$

3. Compute step size $\alpha^{[k]}$ to satisfy

$$\ell_1(z^{[k]} + \alpha^{[k]} d^{[k]}, \eta^{[k]}) \leq \ell_1(z^{[k]}, \eta^{[k]}) + \sigma \alpha^{[k]} \nabla_z \ell_1(z^{[k]}, \eta^{[k]})$$

4. Set

$$z^{[k+1]} := z^{[k]} + \alpha^{[k]} d^{[k]} \qquad \text{and} \qquad \lambda^{[k+1]} := \lambda^{[k]}$$

5. Set $k := k + 1$

Note that the Hessian needs to be symmetric positive definite for Algorithm 3.7 in order to work. The latter can be achieved by utilizing BFGS updates instead of computing the Hessian, cf. [4, 9] for details.

3.2 Interior Point Method

In contrast to the (SQP) approach, the interior point method (IP) is based on constructing approximated solutions, which are strictly contained in the interior of the feasible set $\mathcal{F}$. Hence, each iterate of the interior point algorithm is feasible, quite in contrast to the (SQP) algorithm. This behavior is achieved by attaching penalty terms, which penalize points lying on the boundary of $\mathcal{F}$. Note that this is different from penalty methods discussed in Chapter 2,

which penalize unfeasible points only, i.e. points outside the boundary of $\mathcal{F}$. The method is rather popular by now as one can show that — in contrast to the Simplex method — interior point methods can solve linear optimization problems polynomially regarding the dimension of the problem.

3.2.1 Linear Optimization Problem

Here, we start with the linear case and recall the standard problem of linear optimization in primal normal form

$$\begin{aligned} &\text{minimize} \quad c^\top z \\ &\text{subject to } Az = b, \\ &\qquad\qquad z \geq 0. \end{aligned} \tag{LP}$$

where $A \in \mathbb{R}^{n_H \times n_z}$ represents the linear constraint function, $b \in \mathbb{R}^{n_H}$ the right hand side of the constraints, and $c \in \mathbb{R}^{n_z}$ the cost vector. Utilizing the Lagrangian

$$L(z, \lambda, \mu) = c^\top z + \lambda^\top (-z) + \mu^\top (b - Az)$$

we obtain the KKT conditions

$$A^\top \mu + \lambda = c \tag{3.9}$$
$$Az = b \tag{3.10}$$
$$z \geq 0 \tag{3.11}$$
$$\lambda \geq 0 \tag{3.12}$$
$$\lambda_i z_i = 0 \quad i = 1, \ldots, n_z. \tag{3.13}$$

Now we eliminate the inequalities $z \geq 0$ in the primal problem by including them as penalties in the costs. For $\eta > 0$ we obtain the (logarithmic) barrier problem

$$\begin{aligned} &\text{minimize} \quad c^\top z - \eta \textstyle\sum_{i=1}^{n_z} \log(z_i) \\ &\text{subject to } Az = b. \end{aligned} \tag{BP}$$

Note that $\log(z_i) \to -\infty$ as $z_i \searrow 0$. Hence, the term $-\eta \log(z_i)$ generates a barrier with value ∞ at $z_i = 0$ such that the minimum never lies on the barrier. Now, the aim is to iteratively adapt the parameter η to generate a sequence of feasible solutions $z > 0$, which converges to the minimum of (LP).

Due to the logarithmic terms, the barrier problem (BP) is a nonlinear convex optimization problem. The KKT conditions read

$$c_i - \frac{\eta}{z_i} - (A^\top \mu)_i = 0, \quad i = 1, \ldots, n_z$$
$$Az = b.$$

By defining $\lambda_i := \eta / z_i$, $i = 1, \ldots, n_z$, we can reformulate the KKT conditions to

$$A^\top \mu + \lambda = c \tag{3.14}$$
$$Az = b \tag{3.15}$$
$$\lambda_i z_i = \eta, \quad i = 1, \ldots, n_z. \tag{3.16}$$

Comparing (3.9)–(3.13) to (3.14)–(3.16), we see that the KKT conditions of the barrier problem (BP) can be interpreted as disturbed KKT conditions of (LP) if additionally $z > 0$ and $\lambda > 0$ holds. The disturbance occurs explicitly by the presence of the weighting factor $\eta > 0$ in the complementarity condition (3.13), which then reads (3.16).
If for each $\eta > 0$ the nonlinear equation system (3.14)–(3.16) possesses a solution

$$(z(\eta), \lambda(\eta), \mu(\eta)),$$

then there is hope that this solution converges to the solution of (LP) for $\eta \searrow 0$. The set

$$\{(z(\eta), \lambda(\eta), \mu(\eta)) \mid \eta > 0\}$$

is referred to as *central path*. Since the KKT conditions (3.14)–(3.16) are necessary and due to convexity of problem (BP) also sufficient, the following result holds:

Theorem 3.8
Suppose $\eta > 0$. Then barrier problem (BP) has a solution $z > 0$ if and only if the central path conditions (3.14)–(3.16) have a solution $(z(\eta), \lambda(\eta), \mu(\eta))$ with $z(\eta) > 0$ and $\lambda(\eta) > 0$.

3.2.2 IP Algorithm

To solve (3.14)–(3.16) numerically, Newton's method can be applied to the function

$$F_\eta(z, \mu, \lambda) := \begin{pmatrix} A^\top \mu + \lambda - c \\ Az - b \\ Z\Lambda e - \eta e \end{pmatrix}$$

where

$$Z = \operatorname{diag}(z_1, \ldots, z_{n_z}), \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_{n_z}) \quad \text{and} \quad e = (1, \ldots, 1)^\top.$$

The Jacobian of F_η is given by

$$DF_\eta(z, \mu, \lambda) = \begin{pmatrix} 0 & A^\top & \mathrm{Id} \\ A & 0 & 0 \\ \Lambda & 0 & Z \end{pmatrix}.$$

For this matrix, the following result holds, which will allow us to apply Newton's method:

Theorem 3.9
Suppose $(z, \mu, \lambda) \in \mathbb{R}^{n_z \times n_H \times n_z}$ is a vector with $z > 0$ and $\lambda > 0$ and we have rank$(A) = n_H$. Then the Jacobian $DF_\eta(z, \mu, \lambda)$ is invertible for each $\eta > 0$.

Suppose $(z^{[k]}, \mu^{[k]}, \lambda^{[k]})$ is a given iterate in the Newton method. The Newton correction is given by the linear equation system

$$DF_{\eta^{[k]}}(z^{[k]}, \mu^{[k]}, \lambda^{[k]}) \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \\ \Delta \lambda^{[k]} \end{pmatrix} = -F_{\eta^{[k]}}(z^{[k]}, \mu^{[k]}, \lambda^{[k]})$$

which gives us

$$\begin{pmatrix} 0 & A^\top & \mathrm{Id} \\ A & 0 & 0 \\ \Lambda^{[k]} & 0 & Z^{[k]} \end{pmatrix} \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \\ \Delta \lambda^{[k]} \end{pmatrix} = - \begin{pmatrix} A^\top \mu^{[k]} + \lambda^{[k]} - c \\ Az^{[k]} - b \\ Z^{[k]} \Lambda^{[k]} e - \eta^{[k]} e \end{pmatrix} \tag{3.17}$$

The damped Newton method reveals the new iterate

$$\begin{pmatrix} z^{[k+1]} \\ \mu^{[k+1]} \\ \lambda^{[k+1]} \end{pmatrix} := \begin{pmatrix} z^{[k]} \\ \mu^{[k]} \\ \lambda^{[k]} \end{pmatrix} + \alpha^{[k]} \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \\ \Delta \lambda^{[k]} \end{pmatrix}$$

with step length $\alpha^{[k]} > 0$.
For both the damped and the undamped Newton sequence, one can show that if a central path starts feasible, it will always remain feasible, i.e. if conditions (3.14)–(3.15) hold for $(z^{[0]}, \mu^{[0]}, \lambda^{[0]})$, then the hold for all $(z^{[k]}, \mu^{[k]}, \lambda^{[k]})$, $k > 0$. To guarantee this property, we require

$$\begin{aligned} A^\top \mu^{[k]} + \lambda^{[k]} - c &= 0 \\ Az^{[k]} - b &= 0. \end{aligned}$$

With regards to the Newton iteration (3.17), it follows that

$$\begin{aligned} A^\top \Delta \mu^{[k]} + \Delta \lambda^{[k]} &= 0 \\ A \Delta z^{[k]} &= 0. \end{aligned}$$

For the next iterate, we obtain

$$\begin{aligned} A^\top \mu^{[k+1]} + \lambda^{[k+1]} - c &= A^\top (\mu^{[k]} + \alpha^{[k]} \Delta \mu^{[k]}) + \lambda^{[k]} + \alpha^{[k]} \Delta \lambda^{[k]} - c \\ &= A^\top \mu^{[k]} + \lambda^{[k]} - c \\ &= 0, \\ Az^{[k+1]} - b &= A(z^{[k]} + \alpha^{[k]} \Delta z^{[k]}) - b \\ &= 0, \end{aligned}$$

showing the assertion. Combined, we obtain the following algorithm:

Algorithm 3.10 (Interior Point Method)
Suppose $\varepsilon > 0$, $z^{[0]} \in \mathbb{R}^{n_z}$, $\mu^{[0]} \in \mathbb{R}^{n_H}$, $\lambda^{[0]} \in \mathbb{R}^{n_z}$ to be given and satisfy

$$Az^{[0]} = b, \quad A^\top \mu^{[0]} + \lambda^{[0]} = c, \quad z^{[0]} > 0, \quad \lambda^{[0]} > 0,$$

and set $k = 0$.
While $\frac{z^{[k]\top} \lambda^{[k]}}{n_z} > \varepsilon$ do

1. Set $\sigma^{[k]} \in [0, 1]$ and solve the linear equation system

$$\begin{pmatrix} 0 & A^\top & \mathrm{Id} \\ A & 0 & 0 \\ \Lambda^{[k]} & 0 & Z^{[k]} \end{pmatrix} \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \\ \Delta \lambda^{[k]} \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ Z^{[k]} \Lambda^{[k]} e - \sigma^{[k]} \frac{z^{[k]\top} \lambda^{[k]}}{n_z} e \end{pmatrix}$$

2. Set

$$\begin{pmatrix} z^{[k+1]} \\ \mu^{[k+1]} \\ \lambda^{[k+1]} \end{pmatrix} := \begin{pmatrix} z^{[k]} \\ \mu^{[k]} \\ \lambda^{[k]} \end{pmatrix} + \alpha^{[k]} \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \\ \Delta \lambda^{[k]} \end{pmatrix}$$

where $\alpha^{[k]} > 0$ is chose such that $z^{[k+1]} > 0$ and $\lambda^{[k+1]} > 0$ hold

3. Set $k := k + 1$

Remark 3.11

- *The iterates $z^{[k]}$ are primal feasible since by construction we have*

$$Az^{[k]} = b, \quad z^{[k]} > 0.$$

- *The algorithms can always be executed if we can guarantee rank$(A) = n_H$, cf. Theorem 3.9.*
- *Among conditions (3.9)–(3.13) the complementarity condition $\lambda_i^{[k]} z_i^{[k]} = 0, i = 1, \ldots, n_z$ is not satisfied by the iterates. To meet this condition, we approach it by utilizing the breaking criterion $\frac{z^{[k]\top} \lambda^{[k]}}{n_z}$ also in the iteration.*
- *The algorithm still contains two degrees of freedom, the step size $\alpha^{[k]} > 0$ and the centering parameter $\sigma^{[k]} > 0$. Note that the term $\sigma^{[k]} \frac{z^{[k]\top} \lambda^{[k]}}{n_z}$ plays the role of the penalty parameter η. Depending on the choice of these parameters, we obtain different methods.*

For Algorithm 3.10, we can show that an ε optimal solution can be computed in polynomial time:

Theorem 3.12 (Convergence Interior Point Method)
Suppose $\varepsilon \in (0, 1)$ to be given and $\{(z^{[k]}, \mu^{[k]}, \lambda^{[k]})\}_{k \in \mathbb{N}_0}$ be defined by Algorithm 3.10. Suppose

$$\frac{z^{[k+1]\top} \lambda^{[k+1]}}{n_z} \leq \left(1 - \frac{\delta}{n_z^s}\right) \frac{z^{[k]\top} \lambda^{[k]}}{n_z} \tag{3.18}$$

to hold for parameters $\delta > 0$ and $s > 0$. Moreover, the starting vector $(z^{[0]}, \mu^{[0]}, \lambda^{[0]})$ shall satisfy

$$\frac{z^{[0]\top} \lambda^{[0]}}{n_z} \leq \frac{1}{\varepsilon^\kappa}, \quad \kappa > 0.$$

Then there exists an index $K \in \mathbb{N}$ with $K = \mathcal{O}(n_z^s |\log(\varepsilon)|)$ and $\frac{z^{[k]\top} \lambda^{[k]}}{n_z} \leq \varepsilon$ for all $k > K$.

In implementations, the step size $\alpha^{[k]}$ is typically chosen such that the iterates remain close to the central path. These methods are called *path following methods*. There are feasible and infeasible methods, which either keep the iterates within the feasible set throughout the iteration, or allow violations of the feasible set. The feasible methods are based on smaller sets, which allow for smaller steps only. They are also referred to as *short step methods*, in contrast to infeasible methods which are known as *long step methods*. While converging slower, short step methods still show better convergence properties.

3.2.3 Nonlinear Optimization Problem

After considering the linear case, we now turn towards the nonlinear case and our standard problem

$$\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{with respect to } z \in \mathbb{R}^{n_z} \\ &\text{subject to } G_i(z) \leq 0 \text{ for all } i \in \mathcal{I} \text{ and } H_i(z) = 0 \text{ for all } i \in \mathcal{E} \end{aligned} \tag{NLP}$$

from Definition 1.1. Similar to the linear case, we eliminate the inequality constraints by introducing slack variables $s > 0$ and attaching the restriction to the cost function using logarithmic terms with weighting factors. Note that we want to keep the solution within the feasible set, hence the slack is strictly positive. This gives us the nonlinear barrier problem

$$\begin{aligned} &\text{minimize} \quad F(z) - \eta \sum_{i=1}^{n_G} \log(s_i) \\ &\text{with respect to } (z, s) \in \mathbb{R}^{n_z \times n_G} \\ &\text{subject to } G_i(z) + s_i = 0 \text{ for all } i \in \mathcal{I} \text{ and } H_i(z) = 0 \text{ for all } i \in \mathcal{E} \end{aligned} \tag{NBP}$$

The Lagrangian of the barrier problem reads

$$L(z, s, \lambda, \mu) = F(z) - \eta \sum_{i=1}^{n_G} \log(s_i) + \sum_{i=1}^{n_G} \lambda_i (G_i(z) + s_i) + \sum_{i=1}^{n_H} \mu H_i(z)$$

With $S = \operatorname{diag}(s_1, \ldots, s_{n_G})$, the KKT conditions of the barrier problem (NBP) are given by

$$0 = \nabla_z L(z, s, \lambda, \mu) = \nabla_z F(z) + \sum_{i=1}^{n_G} \lambda_i \nabla_z G_i(z) + \sum_{i=1}^{n_H} \mu \nabla_z H_i(z) \tag{3.19}$$

$$0 = \nabla_s L(z, s, \lambda, \mu) = -\eta S^{-1} e + \lambda \tag{3.20}$$

$$0 = G_i(z) + s_i \qquad i = 1, \ldots, n_G \tag{3.21}$$

$$0 = H_i(z) \qquad i = 1, \ldots, n_H \tag{3.22}$$

We can reformulate (3.20) equivalently into

$$\eta e = S\lambda \qquad \Longleftrightarrow \qquad \eta = s_i \lambda_i, \quad i = 1, \ldots, n_G.$$

Combined with $s_i = -G_i(z) < 0$, $i = 1, \ldots, n_G$ from (3.21), we obtain the so called *disturbed complementarity condition*

$$-\eta = \lambda_i G_i(z), \quad i = 1, \ldots, n_G. \tag{3.23}$$

Similar to the linear case, we aim to guarantee $s_i > 0$ within the Interior Point Method for the nonlinear case. Due to (3.21), the latter is equivalent to $G_i(z) < 0$. Since furthermore (3.23) and $-\eta < 0$ hold, it follows that $\lambda_i > 0$ for $i = 1, \ldots, n_G$. The solution is again characterized by the penalty parameter η and, given that $(z(\eta), s(\eta), \lambda(\eta), \mu(\eta))$ exists, the parametrized solutions again defines the central path.

Similar to the linear case, the nonlinear equation system (3.19)–(3.22) can be solved for $(z(\eta), s(\eta), \lambda(\eta), \mu(\eta))$ via Newton's method. Here, we can follow the steps of the Lagrange–Newton method from Algorithm 3.1. Note that within the nonlinear barrier problem (NBP) we only have equality constraints. Hence, applying the Lagrange–Newton method from Algorithm 3.1 to (NBP) is identical to applying the local SQP method from Algorithm 3.3. Therefore, the search directions can be computed by solving the quadratic approximation (QPE).

Chapter 4

Integer Optimization and Heuristics

In the previous chapters, we assumed that the optimization variable can take any real value within the feasible set $\mathcal{F}$. Within the present chapter, we change this policy and suppose that there exist optimization values, which are from the set of integers only. Utilizing integer variables, we can model logical values, alternatives from finite sets, gear shifts, lot sizes, staff requirements and so on. Since only finitely many integer combinations are possible for a compact set, optimization problems with integer variables appear to be simpler than problems with real variables on first sight. Unfortunately, our basic auxiliary mean of computing an optimum — the gradient — is not available for integer variables. Hence, gradient based methods are not suitable in our present case and we require an alternative. The basis of this chapter is given by the books [1, 8, 9], which we refer to for further details and proofs.

4.1 Mixed Integer Optimization

In contrast to your standard problem in nonlinear optimization (NLP) from Definition 1.1, mixed integer problems additionally contain constraints of type

$$z_i \in \mathbb{Z}.$$

Writing all constraints as implicit constraints, we obtain the following problem:

Definition 4.1 (Mixed Integer Optimization Problem)
We call the problem

$$\begin{aligned} &\text{minimize} && F(z) \\ &\text{with respect to} && z \in \mathbb{R}^{n_{z_1}} \times \mathbb{Z}^{n_{z_2}} \\ &\text{subject to} && z \in \mathcal{F}. \end{aligned} \tag{MINLP}$$

with map $F : \mathbb{R}^{n_{z_1}} \times \mathbb{Z}^{n_{z_2}} \to \mathbb{R}$ and feasible set $\mathcal{F} \subseteq \mathbb{R}^{n_{z_1}} \times \mathbb{Z}^{n_{z_2}}$ a *mixed integer optimization problem in standard form.*

For $n_{z_2} = 0$ we obtain our standard nonlinear optimization problem (NLP), and for $n_{z_1} = 0$ the problem will be purely integer. In case of $z_i \in \mathbb{Z}$ where we only allow for values in $\{0, 1\}$, the variable is called a *boolean* and (MINLP) is referred to as a combinatoric optimization problem. To integrate booleans within problem (MINLP), we add restrictions of the form

$$0 \leq z_i \leq 1.$$

For further details on combinatoric optimization problems can be found in [7].

4.1.1 Mixed Integer Linear Optimization

Here, we start with the linear case with pure integer optimization variable, which is given by

$$\begin{aligned} &\text{minimize} && c^\top z \\ &\text{subject to} && Az = b, \\ &&& z \geq 0, \\ &&& z \in \mathbb{Z}^{n_{z_2}}. \end{aligned} \tag{MILP}$$

where again $A \in \mathbb{R}^{n_H \times n_{z_2}}$ represents the linear constraint function, $b \in \mathbb{R}^{n_H}$ the right hand side of the constraints, and $c \in \mathbb{R}^{n_{z_2}}$ the cost vector.

To motivate why we cannot simply round the real valued solution to the next integer vector, consider the following example:

Example 4.2
Consider the mixed integer linear problem

$$\begin{aligned} \textit{minimize} \quad & -2z_1 - 3z_2 \\ \textit{subject to} \quad & z_1 + 2z_2 \leq 8, \\ & 2z_1 + z_2 \leq 9, \\ & z_1, z_2 \geq 0, \\ & z_1, z_2 \in \mathbb{Z}. \end{aligned}$$

The optimal solution is given by the point $z^\star = (2,3)^\top$ with costs -13. The point $z_{relaxed} = (\frac{10}{3}, \frac{7}{3})^\top$ with costs $-13\frac{2}{3}$ is the optimal solution of the real valued linear optimization problem, where $z_1, z_2 \in \mathbb{Z}$ is replaced by $z_1, z_2 \in \mathbb{R}$. The resulting problem is called relaxation of the integer problem. If we consider the point closest to the relaxed solution $z^{relaxed}$ given by $z = (3,2)^\top$, then we obtain costs of -12, which is not optimal.

From this example we can see that simple rounding to the next feasible integer will not do the trick in general. The relaxation of (MILP) is given by problem (LP), which we considered in Section 3.2.1. This problem possesses a solution, which is quite close to the optimal integer solution, and we will exploit this property more rigorously. Since the feasible set of the relaxed problem (LP) is larger than the feasible set of (MILP), the optimal solution of the relaxed problem is a lower bound for the costs of (MILP):

Theorem 4.3 (Bound for Costs)
Suppose $z^\star$ is the solution of (MILP) and $z^\star_{relaxed}$ the solution of the corresponding relaxation (LP). Then we have

$$c^\top z^\star_{relaxed} \leq c^\top z^\star.$$

There are noteworthy special cases, where the lower bound of the relaxed problem is actually tight, i.e. the relaxed solution is equal to the integer solution $z^\star_{\text{relaxed}} \equiv z^\star$.

Theorem 4.4 (Tight Bound for Costs)
Suppose $z^\star$ is the solution of (MILP) and $z^\star_{relaxed}$ the solution of the corresponding relaxation (LP). If A is totally unimodular, i.e. the determinant of each quadratic submatrix of A is 0, -1 or 1 only, and b is integer, then we have

$$c^\top z^\star_{relaxed} = c^\top z^\star.$$

Totally unimodular constraint matrices occur in many network optimization problems like

- transport problems,
- shortest path problems or
- maximal flow problems.

In this case, any method computing able to exactly compute optimal corners of the feasible set (like the Simplex method) can be applied to solve the integer problem (MILP).

Remark 4.5
Note that more than one corner can be optimal. Hence, Theorem 4.4 compares costs only.

4.1.2 Cutting Plane Method by Gomory

Cutting planes is one method to rigorously utilize the property of closeness of the relaxed solution to the integer solution of (MILP). The idea is to iteratively refine the feasible set. This method is not restricted to the linear case, but may also be applied to (not necessarily differentiable) convex optimization problems. The class of cutting plane methods goes back to works of Gomory in the late 1950s/early 1960s.

Within the cutting plane algorithm, there are three main steps: First, the relaxed problem is solved. The obtained solution is tested whether it is integer. If it is not, then there exists a linear inequality that separates the optimum from the convex hull of the true feasible set. In the second step, the separation problem of finding such an inequality is solved. The respective inequality is called a *cut*. Last, the computed cut is added to the relaxed problem. Since the current non–integer solution is rendered infeasible for the extended problem by the cut, the process can be iterated until an optimal integer solution is found.

Suppose the solution of (LP) is given by

$$z^\star_{B,\text{relaxed}} = A_B^{-1}b - A_B^{-1}A_N z^\star_{N,\text{relaxed}} = \beta_B - \Gamma_B^N z^\star_{N,\text{relaxed}}$$

with basis index set B and non basis index set N. If $z^\star_{B,\text{relaxed}}$ is integer, then $z^\star_{B,\text{relaxed}} = \beta_B$, $z^\star_{N,\text{relaxed}} = 0$ is optimal for (MILP).

If $z^\star_{B,\text{relaxed}}$ is not integer, there exists $i \in B$ such that $\beta_i \notin \mathbb{Z}$. For $\Gamma_B^N = (\gamma_{ij})_{i\in B, j\in N}$ we have that

$$\beta_i = z^\star_{i,\text{relaxed}} + \sum_{j\in N} \gamma_{ij} z^\star_{j,\text{relaxed}}$$

holds for all $i \in B$. Due to $\lfloor \gamma_{ij} \rfloor \leq \gamma_{ij}$ and $z_j^\star \geq 0$ for all $j \in B \cup N$ the inequality

$$\beta_i \geq z^\star_{i,\text{relaxed}} + \sum_{j \in N} \lfloor \gamma_{ij} \rfloor z^\star_{j,\text{relaxed}} \tag{4.1}$$

holds. Now, for a feasible point of (MILP) we have

$$z_i^\star + \sum_{j \in N} \lfloor \gamma_{ij} \rfloor z_j^\star \in \mathbb{Z}$$

and due to the integer property the compared to (4.1) tightened constraint

$$\lfloor \beta_i \rfloor \geq z^\star_{i,\text{relaxed}} + \sum_{j \in N} \lfloor \gamma_{ij} \rfloor z_j^\star \tag{4.2}$$

must hold for any solution of (MILP). Hence, using (4.1) and (4.2) each feasible point of (MILP) must satisfy the relative condition

$$\lfloor \beta_i \rfloor - \beta_i \geq \sum_{j \in N} \left(\lfloor \gamma_{ij} \rfloor - \gamma_{ij} \right) z_j^\star. \tag{4.3}$$

Since $\lfloor \beta_i \rfloor - \beta_i < 0$ for each optimal solution of (LP) with $\beta_i \notin \mathbb{Z}$, $z^\star_{j,\text{relaxed}} = 0$, $j \in N$, this condition can be used to cut the optimal solution of the relaxed problem (MILP).

Combined, we obtain the following algorithm:

Algorithm 4.6 (Cutting Plane Algorithm)
Suppose $z^{[0]} \in \mathbb{R}^{n_{z_2}} \setminus \mathbb{Z}^{n_{z_2}}$ to be given and set $k = 0$.
While $z^{[k]} \notin \mathbb{Z}^{n_{z_2}}$ do

1. Solve the relaxed problem (LP) of problem (MILP) to obtain the sets B and N as well as the solution $z^\star_{B,\text{relaxed}} = \beta_B$ and $z^\star_{N,\text{relaxed}}$
2. If the (LP) has no solution, then (MILP) has no solution, STOP
3. Fix index $i \in B$ satisfying $\beta_i \notin \mathbb{Z}$ and construct the cut

$$z_{n_{z_2}+1} + \sum_{j \in N} \left(\lfloor \gamma_{ij} \rfloor - \gamma_{ij} \right) z_j^\star = \lfloor \beta_i \rfloor - \beta_i$$
$$z_{n_{z_2}+1} \geq 0$$

   where $z_{n_{z_2}+1}$ is a slack variable for (4.3)
4. Add the cut to problem (MILP) to obtain (MILP)′ and set (MILP) := (MILP)′

While being simple in construction, each iteration adds one constraint and one slack variable to the problem. Since the growing number leads to increasing computing times of the steps, this is the main disadvantage of Algorithm 4.6. An efficient implementation relies on the dual simplex algorithm and is beyond the scope of the lecture. For details on the latter and finite termination of Algorithm 4.6, we refer to [8] and references therein.

The cutting plane method can also be applied to nonlinear problem (MINLP), yet it requires the feasible set $\mathcal{F}$ to be convex. To obtain a respective algorithm, one simply replaces (LP) by (NLP) in Step 1 of Algorithm 4.6.

4.1.3 Branch and Bound Method

One of the popular methods to solve mixed integer problem of linear and nonlinear type is the so called *Branch and Bound Method*, sometimes also termed *Branch and Cut Method*. As the name already states, the method incorporates two ideas: branching and bounding/cutting.

The branch idea generates a systematic enumeration of candidate solutions by means of a state space search using a tree structure. The root of this tree is given by the full set of choices for all variables. On the next level of the tree, one variable is branched out, i.e. for each possible choice of the variable a subproblem is generated within which the variable is fixed. Applying this branching idea to all variables, we obtain a tree where the variables of problems on the last level, the so called leafs, are completely fixed. Hence, either the feasible set is empty or contains exactly one element, rendering the solution for each leaf to be easily obtainable. Unfortunately, for large number of variables and choices, the number of leafs grows rapidly.

To avoid checking all leafs, the bounding idea is used to rule out branches completely without having to consider all their leafs. To this end, the branch is checked against bounds on the optimal solution, and is discarded if it cannot produce a better solution than the best one found by the algorithm so far. There are several situations where analyzing subtrees/branches doesn't make sense anymore:

- **Infeasibility:** Fixing variables makes the feasible set smaller. Hence, if the feasible set of a node is empty, then the feasible sets of its branches are empty as well. These branches can be cut.

- **Optimal Nodes:** If the optimization problem of a node can be solved, it already incorporates the optimal solution among the solutions of its branches. Hence, we can omit solving all its branches.

- **Bounding:** Considering two nodes on the same level, we can estimate the lower and upper bounds on the solution of each node. If these ranges are intersection free, the one of them and all its branches can be ruled out.

These bounding/cutting rules can help to drastically reduce the size of the search tree. The performance of the cuts depends on the quality of the bounds, for which the following approaches are useful:

- Each feasible point reveals an upper bound for the optimal value of (MINLP).

- Each optimal solution of the relaxation (NLP) gives us an upper bound for the optimal value of (MINLP).

- Each feasible solution of a suitable chosen dual problem presents a lower bound for the optimal value of (MINLP).

Combining these ideas, a general Branch and Bound Method consists of the following rules:

1. The branching rule defines how new nodes in the tree are generated. The rule determines how the feasible set is partitioned.

2. The bounding rule defines lower and upper bounds for the optimal value.

3. The traversing rule determines the sequence of nodes being handled, i.e. depth–first or breadth–first search.

4. The cutting rule assesses when computing the solution of a node and respective branches can be omitted.

We now apply these rule to the linear case (LP). The nonlinear case can be treated similarly.

Branching

Consider a node within the branch and bound search tree to be given by problem

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{Z}^{n_{z_2}}.} \tag{MILP}$$

Suppose $z^\star$ is the solution of (MILP) and $z^\star_{\text{relaxed}}$ the solution of the relaxed problem

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{R}^{n_{z_2}}.} \tag{LP}$$

If $z^\star_{\text{relaxed}}$ is feasible for (MILP), then no further branching is required. Otherwise we have $z^\star_{\text{relaxed}} \notin \mathbb{Z}^{n_{z_2}}$, and branching consist of the following two steps:

(i) Choose an index k with $z^\star_{k,\text{relaxed}} \notin \mathbb{Z}$

(ii) Generate left and right nodes (S_L) and (S_R) given by the problems

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{Z}^{n_{z_2}},\ z_k \leq \lfloor z^\star_{k,\text{relaxed}} \rfloor} \tag{S_L}$$

and

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{Z}^{n_{z_2}},\ z_k \geq \lfloor z^\star_{k,\text{relaxed}} \rfloor + 1} \tag{S_R}$$

Applying this branching rule recursively with (LP) as root, we obtain the branching tree structure. Each edge in this tree represents a new additional constraint, and the union of the feasible sets of (S_L) and (S_R) is equivalent to the feasible set of the parenting node.

Bounding

Again, we consider a single node and the respective mixed integer linear optimization problem

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{Z}^{n_{z_2}}.} \tag{MILP}$$

Then we obtain an upper bound $b_u \in \mathbb{R}$ via

$$c^\top z^\star \leq b_u := c^\top z \qquad \forall z \in \mathcal{F}, \tag{4.4}$$

i.e. each feasible point reveals an upper bound. A lower bound $b_l \in \mathbb{R}$ can be obtained by utilizing the relaxed problem

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{R}^{n_{z_2}}.} \tag{LP}$$

Here, we have that

$$b_l := c^\top z^\star_{\text{relaxed}} \leq c^\top z^\star. \tag{4.5}$$

Alternatively, a lower bound can be computed using any feasible point of dual problem, which can be shown via the Weak Duality Theorem.

Cutting

A node is termed *examined* and any further branching can be omitted if one of the following conditions applies:

- The relaxed problem (LP) is infeasible.
- The relaxed problem (LP) has an integer solution.
- The lower bound of the node is larger or equal to the current upper bound.

Branch and Bound Algorithm

Having discussed the rules of the branch and bound algorithm, we can now combine them to obtain an integrated algorithm:

Algorithm 4.7 (Branch and Bound Algorithm)
Suppose $b_u \in \mathbb{R} \cup \{\infty\}$ is given. Initialize set of active nodes $\mathcal{S} := \{(\text{MILP})\}$.
While $\mathcal{S} \neq \emptyset$ do

1. Choose node $S \in \mathcal{S}$ according to traversing rule
2. Compute optimal solution $z^\star_{\text{relaxed}}$ of the relaxed problem of S
3. If $z^\star_{\text{relaxed}}$ is bounded, set $\overline{b} := c^\top z^\star_{\text{relaxed}}$
   Else set $\overline{b} := -\infty$
4. If S is infeasible, remove S from $\mathcal{S}$
5. If $z^\star_{\text{relaxed}} \in \mathbb{Z}^{n_{z_2}}$ then
   (a) If $\overline{b} < b_u$ then
   Save $z^\star_{\text{relaxed}}$ as current best solution
   Set $b_u := \overline{b}$
   (b) Remove S from $\mathcal{S}$
6. If $\overline{b} \geq b_u$ then
   Remove S from $\mathcal{S}$
7. If $\overline{b} < b_u$ then
   (a) Choose an index k with $z^\star_{k,\text{relaxed}} \notin \mathbb{Z}$
   (b) Generate left and right nodes (S_L) and (S_R) of S given by the problems

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{Z}^{n_{z_2}},\ z_k \leq \lfloor z^\star_{k,\text{relaxed}} \rfloor} \qquad (S_L)$$

   and

$$\boxed{\text{minimize } c^\top z \quad \text{subject to } Az = b,\ z \geq 0,\ z \in \mathbb{Z}^{n_{z_2}},\ z_k \geq \lfloor z^\star_{k,\text{relaxed}} \rfloor + 1} \qquad (S_R)$$

   (c) Set $\mathcal{S} := \mathcal{S} \setminus \{S\} \cup \{(S_L), (S_R)\}$

The size of the tree generated by the branch and bound method depends on the data of the problem and can be very large. In practice, the algorithm may have to be stopped without having found the optimal solution. Yet, even after stopping, upper and lower bounds are available. The upper bound is given by the current b_u whereas the lower bound is given by the minimum over all b_l of active nodes. These values can also be used to terminate the iteration, e.g. via $b_u - b_l < \varepsilon$ with a user defined tolerance $\varepsilon > 0$.

Additionally, the Branch and Bound method can be combined with the Cutting Edge method. This combination allows to improve the quality of the bounds and leads to much smaller search trees.

4.2 Heuristics

The second class of algorithms we consider in the context of mixed integer nonlinear optimization problems (MINLP) are so called heuristics. In contrast to the gradient based methods, heuristics always aim for global minima, while gradient based methods may get stuck in local ones. Moreover, heuristics can be designed to deal even with non–differentiable problems, requiring function evaluations only and avoiding first order information, which makes them not only easily applicable, but they also cover a broader class of problems. The denotation *heuristics* is due to the fact that for the methods no convergence results exists. Hence, we cannot guarantee that a method will return a local or global minimum or even only a stationary point. Despite these shortcomings, the methods have shown to provide good results in practice.

4.2.1 Nelder Mead Algorithm

Since the Nelder Mead algorithm is independent from requirements of the cost function F, it has become quite popular. The method is based on the construction of simplexes.

Definition 4.8 (Simplex)
Consider vectors $z^0, \dots, z^n \in \mathbb{R}^{n_z}$ where the differences $z^i - z^0$, $i = 1, \dots, n$ are linearly independent. The convex hull of these points

$$\mathcal{S} = \left\{ z = \sum_{i=0}^{n} \lambda_i z^i \mid \lambda_i \geq 0,\ i = 0, \dots, n,\ \sum_{i=0}^{n} \lambda_i = 1 \right\} \tag{4.6}$$

is called n-dimensional simplex with vertexes $z^0, \dots, z^n$.

Upon start of the method, a simplex $\mathcal{S}^{[0]}$ is given. Now, each iterate of the method consists of the following steps:

1. For the current simplex $\mathcal{S}^{[k]}$ with vertexes $z^0, \dots, z^n$ identify the vertex z^m with maximal cost function value

$$F(z^m) = \max\{F(z^0), \dots, F(z^n)\}.$$

2. Compute a point $\tilde{z}$ satisfying $F(\tilde{z}) < F(z^m)$ and replace z^m by $\tilde{z}$ to obtain the new simplex $\mathcal{S}^{[k+1]}$ with vertexes $\tilde{z}$ and z^i, $i \neq m$.

For the following computations, we utilize the center of mass of the vertexes regarding z^j, which is given by

$$c^j = \frac{1}{n} \sum_{\substack{i=0 \\ k \neq j}}^{n} z^i.$$

To compute the new point $\tilde{z}$ in the second step of the outlined procedure, we apply three construction principles:

(i) **Reflection:** The new point z^r is given by reflection of vertex z^m at center of mass c^m

$$z^r := c^m + \gamma(c^m - z^m), \quad 0 < \gamma \leq 1.$$

(ii) **Expansion:** The new point z^e is pushed out further in direction $c^m - z^m$

$$z^e := c^m + \beta(z^r - c^m), \quad \beta > 1.$$

(iii) **Contraction:**

- **Inner partial contraction:** The point z^c is shifted between z^m and c^m

$$z^c := c^m + \alpha(z^m - c^m), \quad 0 < \alpha \leq 1.$$

- **External partial contraction:** The point z^c is shifted between z^r and c^m

$$z^c := c^m + \alpha(z^r - c^m), \quad 0 < \alpha \leq 1.$$

- **Total contraction:** The points z^i, $i = 0, \ldots, n$, $i \neq k$ are replaced by the centers of the segments between z^i and z^k

$$\hat{z}^i := z^i + \frac{1}{2}(z^k - z^i) = \frac{1}{2}(z^i + z^k).$$

Combined, we obtain the following algorithm:

Algorithm 4.9 (Nelder Mead Algorithm)
Suppose initial value $z^{[0,0]} \in \mathbb{R}^{n_z}$, and parameters $0 < \alpha < 1$, $\beta > 1$ and $0 < \gamma \leq 1$ as well as the tolerance $\text{tol} > 0$ to be given and set $k := 0$.

- Set the vertexes of the initial simplex $\mathcal{S}^{[0]}$

$$z^{[(0,i)]} = z^{[0,0]} + e_i$$

where e_i denotes the i-th unity vector

- While $\left(\frac{1}{n+1} \sum_{i=0}^{n} \left[F(z^{[k,i]}) - \frac{1}{n+1} \sum_{i=0}^{n} F(z^{[k,i]}) \right]^2 \right)^{1/2} > \text{tol}$ do

1. Determine a vertex $z^{[(k,m)]}$ satisfying

$$F(z^{[(k,m)]}) = \max\{F(z^{[(k,0)]}), \ldots, F(z^{[(k,n)]})\}$$

and a vertex $z^{[(k,l)]}$ satisfying

$$F(z^{[(k,l)]}) = \min\{F(z^{[(k,0)]}), \ldots, F(z^{[(k,n)]})\}$$

2. Compute the center of mass with respect to $z^{[(k,m)]}$

$$c^m = \frac{1}{n} \sum_{\substack{i=0 \\ i \neq m}}^{n} z^{[(k,i)]}.$$

3. Compute the reflection point of $z^{[(k,m)]}$ at c^m

$$z^r := c^m + \gamma(c^m - z^{[(k,m)]})$$

   (a) If $F(z^r) < F(z^{[(k,l)]})$, then z^r is the new minimal point. Test whether the expansion point

$$z^e := c^m + \beta(z^r - c^m)$$

   improve the cost function even further, set

$$z^{[(k+1,m)]} = \begin{cases} z^e, & \text{if } F(z^e) < F(z^r) \\ z^r & \text{if } F(z^e) \geq F(z^r) \end{cases} \quad \text{and} \quad z^{[(k+1,i)]} = z^{[(k,i)]} \; \forall i \neq m$$

   and goto 4

   (b) If $F(z^{[(k,l)]}) \leq F(z^r) \leq \max\{F(z^{[(k,i)]}) \mid i \notin \{l, m\}\}$, then set

$$z^{[(k+1,m)]} = z^r \quad \text{and} \quad z^{[(k+1,i)]} = z^{[(k,i)]} \; \forall i \neq m$$

   and goto 4. Here, z^r is at most worse than $z^{[(k,l)]}$ and normally better than $z^{[(k,m)]}$

   (c) If $F(z^r) > \max\{F(z^{[(k,i)]}) \mid i \neq m\}$, then

      i. If $F(z^r) > F(z^{[(k,m)]})$, then z^r is a deterioration and it is typically advisable not to leave the simplex. Apply inner partial contraction and compute

$$z^c := c^m + \alpha(z^{[(k,m)]} - c^m)$$

      ii. If $F(z^r) \leq F(z^{[(k,m)]})$, then z^r improves $z^{[(k,m)]}$, yet it is worse than other vertexes. It is typically advisable to search closer to the simplex. Apply outer partial contraction and compute

$$z^c := c^m + \alpha(z^r - c^m)$$

   (d) If $F(z^c) < F(z^{[(k,m)]})$, then set $z^{[(k+1,m)]} = z^c$
   Else, no improvement could be achieved. Apply a total contraction with respect to the current best point $z^{[(k,l)]}$ and set

$$z^{[(k+1,l)]} := z^{[(k,l)]} \qquad \text{and} \qquad z^{[(k+1,i)]} := \frac{1}{2}(z^{[(k,i)]} + z^{[(k,l)]}) \; \forall i \neq l.$$

4. Set $k := k + 1$

Within Algorithm 4.9, we utilized the standard deviation of the cost function values at the vertexes of the simplex as a termination criterion. Apart from this criterion used, e.g., in the

NAG library, one can also utilize the diameter of the simplex $\mathcal{S}^{[k]}$ to abort the iteration.

The vertex with minimal costs $z^{[k,l]}$ can be interpreted as the current iterate. By construction, we can show

$$F(z^{[k+1,l]}) \leq F(z^{[k,l]}), \qquad k = 0, 1, \ldots$$

Within practical applications, the constants $0.4 \leq \alpha \leq 0.6$, $2 \leq \beta \leq 3$ and $\gamma = 1$ have proven to be of value.

4.2.2 Evolution Algorithm

An evolution algorithm is inspired by mechanisms of biological evolution, such as reproduction, mutation, recombination, and selection. In practice, these algorithm are very widespread, typical applications are operation sequencing, personell planning, container transport, minimization of setup times of machines, nonlinear mixed integer optimization, structure optimization and many more. Candidate solutions to the optimization problem play the role of individuals in a population and are assessed by a fitness function, which determines their quality. Evolution of the population then takes place after the repeated application of the above operators.

Evolutionary algorithms often perform well at approximating solutions to all types of problems because they ideally do not make any assumption about the underlying fitness landscape. Yet, applying evolutionary algorithms typically only shows good initial improvements and finding an optimum with sufficient accuracy is leading to difficulties. Moreover, the algorithm contains stochastic components and no convergence proofs are available. Last, the incorporation of restrictions is problematic and is typically realized via penalty function, cf. Chapter 2

Modeling

Here, we consider the constrained optimization problem

$$\boxed{\begin{aligned} &\text{minimize} \quad F(z) \\ &\text{with respect to } z \in \mathcal{F} \subset \mathbb{R}^{n_z}. \end{aligned}} \tag{PP}$$

where again $\mathcal{F}$ denotes the feasible set of the problem and $F : \mathbb{R}^{n_z} \to \mathbb{R} \cup \{\pm\infty\}$ is an arbitrary function satisfying $F(z) \neq \infty$ for all $z \in \mathcal{F}$. Apart from this restriction, no other assumptions are made regarding the cost function F. The restriction $z \in \mathcal{F}$ can formally be modelled via $F(z) = -\infty$ for all $z \notin \mathcal{F}$ and optimizing over the entire $\mathbb{R}^{n_z}$.

Evolutionary algorithms simulate the natural selection. To formalize respective operators, we introduce the following denotation:

Definition 4.10 (Individual, Fitness Function, Population)
Suppose problem (PP) to be given.

- Each $z \in \mathcal{F}$ is called an *individual*.
- the function $F : \mathbb{R}^{n_z} \to \mathbb{R} \cup \{\pm\infty\}$ is called *fitness function*, its value is called the *fitness* of an individual.
- A subset $\{z_1, \ldots, z_k\} \subset \mathcal{F}$ is called *population* of size k.

The aim of each evolutionary algorithm is to find an individual with maximal fitness. To this end, the following operators are applied:

Definition 4.11 (Mutation, Recombination, Selection)
Suppose problem (PP) to be given.

- A *mutation* operator is a mapping $\mathcal{M} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$, which is subject to a stochastic process.
- A function $\mathcal{R} : (\mathbb{R}^{n_z})^q \to (\mathbb{R}^{n_z})^r$, which combined q individual with each other to generate r new individuals is called a *recombination* operator. The recombination operator may be subject to a stochastic process.
- A function $\mathcal{S} : (\mathbb{R}^{n_z})^s \to (\mathbb{R}^{n_z})^p$ is called a *selection* operator, if it selects p individuals from a set of $s \geq p$ individuals via their respective fitness possibly using a stochastic process.

To illustrate the operators, we consider the following example:

Example 4.12
Suppose $n_z = 1$ and $x, y \in \mathbb{R}$. Then the mappings

- $\mathcal{M} : \mathbb{R} \to \mathbb{R}$ *with* $\mathcal{M}(x) := x + \mathcal{N}(0, 1)$,
- $\mathcal{R} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *with* $\mathcal{R}(x, y) := (x + y)/2$, *and*
- $\mathcal{S} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *with* $\mathcal{S}(x, y) := \operatorname{argmin}\{F(x), F(y)\}$

define mutation, recombination and selection operators. In this example, the mutation operator is subject to a stochastic process, while the recombination and selection operators are deterministic.

Popular Operators

In applications, some operators have become rather popular. Here, we discuss the principles of these operators on a very basic scale. Suppose that $\mathcal{F}$ is a set with finitely many elements. The individuals of this set can be coded as bit strings of length ℓ such that

$$\mathcal{F} \subseteq \{0, 1\}^\ell.$$

1–Point–Crossover Suppose two individuals x and y are given and used to generate a new individual via

$$\left(x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_\ell \end{pmatrix}, \ \sigma \in \{0, \ldots, \ell\}, y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_\ell \end{pmatrix} \right) \mapsto \begin{pmatrix} y_1 \\ \vdots \\ x_\sigma \\ y_{\sigma+1} \\ \vdots \\ y_\ell \end{pmatrix}.$$

Uniform–Crossover Suppose two individuals x and y are given and used to generate a new individual via

$$\left(x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_\ell \end{pmatrix}, \ 1 \leq \sigma_1 < \sigma_2 < \ldots < \sigma_\nu \leq \ell, y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_\ell \end{pmatrix} \right) \mapsto \begin{pmatrix} y_1 \\ \vdots \\ x_{\sigma_1 - 1} \\ y_\sigma \\ x_{\sigma_1 + 1} \\ \vdots \end{pmatrix}.$$

Mutation Suppose p_m is the probability of mutation of a bit. Then an individual x mutates to a new individual v via

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_\ell \end{pmatrix} \mapsto \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_\ell \end{pmatrix} = y,$$

where $\mathrm{P}\left(x_k \neq y_k\right) = p_m$ for all $k = 1, \ldots, \ell$.

Fitness Proportional Selection The probability of selection a individual x_k for the next parent generation is proportional to its fitness. The selection of μ individual is performed by turning a wheel of fortune with one selection arrow μ times, or alternatively to turn it once with μ selection arrows.

Elite Selection The probability of selection a individual x_k is 1 if it exhibits the best fitness.

Combining Operators to Algorithm

To conclude this section, we new combine the presented general forms of mutation, recombination and selection operators to form a unified algorithm. We like to note that by applying a evolutionary algorithm, we cannot guarantee that the algorithm will terminate after a finite number of steps. Indeed, evolutionary algorithms tend to require a large number of iterations, and the stochastic nature of the operators may hinder a replication of results.

Algorithm 4.13 (Concept of an Evolution Algorithm)
Initialize population $\mathcal{P}^{[(0)]}$ with m individuals and set $k := 0$.
While a suitable breaking criterion is not met do

1. Choose q individuals $x_j \in \mathcal{P}^{[(k)]}$, $j = 1, \ldots, q$ and generate new individuals y_j, $j = 1, \ldots, r$ by applying the recombination operator

$$(y_1, \ldots, y_r) := \mathcal{R}(x_1, \ldots, x_q)$$

2. Apply the mutation operator on the recombined individuals

$$z_j := \mathcal{M}(y_j), \qquad j = 1, \ldots, r$$

3. Apply the selection operator on the population $\mathcal{P}^{[(k)]} \cup \{z_1, \ldots, z_r\}$ and set

$$\mathcal{P}^{[(k+1)]} := \mathcal{S}(x_1, \ldots, x_m, z_1, \ldots, z_r)$$

4. Set $k := k + 1$

Part II

Economic Processes

Chapter 5

Production and Inventory

The so called *production* and *inventory problems* belong to the classical topics in Operations Research. Starting from the 1950s, a number of models and methods arose. Here, we discuss some of the deterministic continuous time models and leave aside stochastic processes. The basis of this chapter is the book of Feichtinger and Hartl [2], which we will use without further notice.

First, we describe a model for production and inventory, which is subject to a given demand and nonnegativity conditions from the production rate and the inventory stock. In this regards, we introduce the concept of the decision and prediction horizons. Thereafter, we discuss the simultaneous choice of an optimal production and price policy. Without further notice, we assume all functions used in this chapter to be continuously differentiable in their arguments.

5.1 Production and Inventory for given Demand

Within this section, we suppose the demand $d(t)$ for a product to be given on an interval $[0, T]$. The demand is assumed to be positive and continuously differentiable. To satisfy the demand, the products can either be produced or be taken from the inventory. Here, we denote the production rate by $v(t)$ and the inventory level by $z(t)$. Hence, the inventory rate of change is given by the difference between the production rate and the demand

$$\dot{z}(t) = v(t) - d(t), \quad z(0) = z_0, \tag{5.1}$$

where $z_0 \in \mathbb{R}_0^+$ denotes the inventory level at the beginning of the considered interval $[0, T]$. Moreover, we denote the production and inventory costs by $c_{\text{prod}}(v, t)$ and $c_{\text{inv}}(z, t)$. Upon termination of the planning, the state of the inventory is assessed via the function $c_{\text{inv},T}(z(T), T)$. Therefore, the total costs arising for the production and inventory planning problem are given by

$$J_T(v, z) = \int_0^T \left(c_{\text{prod}}(v(t), t) + c_{\text{inv}}(z(t), t)\right) dt + c_{\text{inv},T}(z(T), T). \tag{5.2}$$

Additionally, the production rate and the inventory level shall be non negative, which reveals the constraints

$$0 \leq v(t) \leq \overline{v}, \quad 0 \leq z(t) \leq \overline{z} \quad \forall t \in [0, T]. \tag{5.3}$$

Note that the system dynamics is given by (5.1), where $z(\cdot)$ is the state of the system, $v(\cdot)$ the external control and $d(t)$ an external known disturbance.

5.2 HMMS Model

One approach to solve the production and inventory problem is the linear quadratic problem introduced by Holt, Modigliani, Muth and Simon in the 1960s, which is referred to as the HMMS model. Within this model, the dynamics are linear differential equations and the costs are quadratic functionals. In particular, the inventory costs are assessed as quadratic deviations from a wanted stock level $\tilde{z}(t)$ and the production costs are penalized regarding their deviation from the ideal production level $\tilde{v}(t)$. Hence, we obtain

$$c_{\text{inv}}(z(t),t) = \frac{1}{2} c_{\text{inv}} \cdot (z(t) - \tilde{z}(t))^2 \tag{5.4}$$

$$c_{\text{prod}}(v(t),t) = \frac{1}{2} (v(t) - \tilde{v}(t))^2 \tag{5.5}$$

$$c_{\text{inv},T}(z(T),T) = \frac{1}{2} c_{\text{inv},T} \cdot (z(T) - \tilde{z}(T))^2 \tag{5.6}$$

Note that we can omit to add a weighting parameter to the second equation (5.5), instead the weighting can be balanced using the parameters from (5.4), (5.6). For this model, the LQ solution approach via the Riccati equations can be followed, which is beyond the scope of this lecture. The optimal solution for problem (5.1), (5.2), (5.4), (5.5), (5.6)

$$\begin{aligned}
&\text{minimize} \quad J_T(v,z) = \frac{1}{2} \int_0^T \left((v(t) - \tilde{v}(t))^2 + c_{\text{inv}} \cdot (z(t) - \tilde{z}(t))^2 \right) dt \\
&\qquad\qquad + \frac{1}{2} c_{\text{inv},T} \cdot (z(T) - \tilde{z}(T))^2 \\
&\text{with respect to } v : [0,T] \to \mathbb{R}_0^+ \\
&\text{subject to } \dot{z}(t) = v(t) - d(t), \quad z(0) = z_0.
\end{aligned}$$

is given by

$$v(t) = \tilde{v}(t) - z(t)\sqrt{c_{\text{inv}}} \tanh\left(\sqrt{c_{\text{inv}}} T + \operatorname{arctanh}(c_{\text{inv},T}/\sqrt{c_{\text{inv}}}) - t\right) + \gamma(t). \tag{5.7}$$

where $\gamma(t)$ is given by the differential equation

$$\begin{aligned}
\dot{\gamma}(t) =& \gamma(t)\sqrt{c_{\text{inv}}} \tanh\left(\sqrt{c_{\text{inv}}} T + \operatorname{arctanh}(c_{\text{inv},T}/\sqrt{c_{\text{inv}}}) - \sqrt{c_{\text{inv}}} t\right) \\
&+ \sqrt{c_{\text{inv}}} \tanh\left(\sqrt{c_{\text{inv}}} T + \operatorname{arctanh}(c_{\text{inv},T}/\sqrt{c_{\text{inv}}}) - \sqrt{c_{\text{inv}}} t\right) (\tilde{v}(t - d(t))) - c_{\text{inv}} \cdot \tilde{z}(t)
\end{aligned}$$

with terminal condition $\gamma(T) = c_{\text{inv},T} \cdot \tilde{z}(T)$. The optimal production rate equals the idea production level corrected by two summands: The first correction term depends on the current stock and reduces the production rate proportionally to the current stock level. The reduction is increasing if either c_{inv} is larger, if the terminal time T is further away or if $c_{\text{inv},T}$ is larger. The second correction term depends on the model parameters. For the plausible case $\tilde{v} \leq d$ for all $t \in [0,T]$ and $c_{\text{inv},T} \geq 0$ we have $\gamma(t) > 0$. An increase in demand then triggers an increase of γ, and in turn of the production rate v.

Remark 5.1
In the special case $\tilde{v} = d$ for all $t \in [0,T]$, $\tilde{z} = const$ and $c_{inv,T} = 0$, we obtain γ, v and z in the closed form

$$\begin{aligned}
\gamma(t) &= \sqrt{c_{inv}} \tilde{z} \tanh\left(\sqrt{c_{inv}}(T-t)\right) \\
v(t) &= d(t) + (\tilde{z} - z(t))\sqrt{c_{inv}} \tanh\left(\sqrt{c_{inv}}(T-t)\right) \\
z(t) &= \tilde{z} + (z_0 - \tilde{z}) \cosh\left(\sqrt{c_{inv}}(T-t)\right) / \cosh\left(\sqrt{c_{inv}} T\right).
\end{aligned}$$

Unfortunately, due to the lack of constraints (5.3), the linear quadratic inventory problem is unrealistic. To include such constraints for the simplest case of linear costs, let us consider that the production and inventory costs neither depend on the time nor on the batch size

$$\begin{aligned}
&\text{minimize} \quad J_T(v,z) = \int_0^T \left(c_{\text{prod}} v(t) + c_{\text{inv}} z(t)\right) dt \\
&\text{with respect to } v : [0,T] \to \mathbb{R}_0^+ \\
&\text{subject to } \dot{z}(t) = v(t) - d(t), \quad z(0) = z_0, \\
&\qquad\qquad 0 \le v(t) \le \overline{v} \text{ and } z(t) \ge 0 \quad \forall t \in [0,T].
\end{aligned}$$

Due to linearity of the control, the production is limited by the maximal rate $\overline{v}$. Here, we assume that $d(t) \le \overline{v}$ holds for all $t \in [0,T]$, i.e. the demand can always be satisfied. Since the terminal inventory is not assessed, we not necessarily have $v < \overline{v}$.

From an economic point of view, it is clear that the optimal strategy possesses the following structure: There is no production until the initial inventory is empty. Thereafter, the production rate and the demand rate coincide. Denoting the accumulated demand by D and defining the time instant t_1 via

$$D(t_1) = \int_0^{t_1} d(t)dt = z_0, \tag{5.8}$$

the optimal solution is given by

$$v(t) = \begin{cases} 0, & 0 \le t < t_1 \\ d(t) & t_1 \le t \le T \end{cases}, \tag{5.9}$$

which gives us

$$z(t) = \begin{cases} z_0 - D(t), & 0 \le t < t_1 \\ 0 & t_1 \le t \le T \end{cases}. \tag{5.10}$$

Note that the optimal strategy remains the same even if general inventory costs $c_{\text{inv}}(z(t),t) > 0$ for $z(t) > 0$ are used. In contrast to the HMMS model, no production smoothing occurs. As soon as the stock is consumed, the production rate suffers from the fluctuations of the demand. This property of the optimal strategy is due to the linearity of the production costs. Applying convex production costs as outlined in the following section, the optimal production is smoothed out.

5.3 Arrow-Karlin-Model

The model from Arrow and Karlin (late 1950s) is the starting point for a series of Inventory and Production models. Within this model, it is assumed that production costs are time independent and marginally increasing. Additionally, for simplicity of exposition, we assume the inventory costs to be linear, which gives us

$$\begin{aligned}&\text{minimize} \quad J_T(v,z) = \int_0^T \left(c_{\text{prod}}(v(t)) + c_{\text{inv}} z(t)\right) dt + c_{\text{inv},T}(z(T),T)\\ &\text{with respect to } v : [0,T] \to \mathbb{R}_0^+\\ &\text{subject to } \dot{z}(t) = v(t) - d(t), \quad z(0) = z_0,\\ &\qquad\qquad 0 \le v(t) \le \overline{v} \text{ and } 0 \le z(t) \le \overline{z} \quad \forall t \in [0,T].\end{aligned}$$

where we have $\dot{c}_{\text{prod}}(v(t)) > 0$ for $v(t) > 0$ and $\ddot{c}_{\text{prod}}(v(t)) > 0$. Similar to the linear case in the previous Section **??**, we suppose that $d(t) \le \overline{v}$ holds for all $t \in [0,T]$, i.e. the demand can always be satisfied.

For this problem, we can directly derive the optimal solution for a very special case:

Lemma 5.2
Suppose the Arrow–Karlin model to be given. If we have $z_0 < D(T)$, then the inventory satisfies $z(T) = 0$. Otherwise, $v(t) = 0$ for all $t \in [0,T]$ is optimal, i.e. nothing is produced.

We now extend this case to so called *boundary solution intervals* and *inner solution intervals*. We define a boundary solution interval $[\tau_1, \tau_2]$ by $z(t) = 0$ for all $t \in [\tau_1, \tau_2]$ where $\tau_1 = 0$ or $z(\tau_1 - \varepsilon) > 0$ and $\tau_2 = T$ or $z(\tau_2 + \varepsilon) > 0$ for small $\varepsilon > 0$. We call $[t_1, t_2]$ an inner solution interval if $z(t) > 0$ for $t \in (t_1, t_2)$, $t_1 = 0$ or $z(t_1) = 0$ and $t_2 = T$ or $z(t_2) = 0$.

For these particular intervals, the following holds:

Lemma 5.3 (Optimal Strategy on Inner Solution Intervals)
On an inner solution interval the production rate satisfies $v(t) > 0$ and we have

$$v(t) = \left(\dot{c}_{prod}(v(t))\right)^{-1} \left(\lambda_0 + c_{inv} t\right) \tag{5.11}$$

for a constant λ_0, which is defined later and different for each inner solution interval.

Lemma 5.4 (Optimal Strategy on Boundary Solution Intervals)
On an boundary solution interval the production rate is equal to the demand

$$v(t) = d(t) > 0 \tag{5.12}$$

and we have

$$c_{inv} \ge \dot{d}(t) \ddot{c}_{prod}(d(t)). \tag{5.13}$$

From Lemma 5.4, we can directly conclude a result similar to the linear constrained case from Section **??**:

Theorem 5.5 (Full Boundary Solution)
If (5.13) hold for all $t \in [0,T]$, then the production rate is identical to the demand, i.e.

$$v(t) = d(t) \tag{5.14}$$
$$z(t) = 0 \tag{5.15}$$

for all $t \in [0,T]$.

Additionally, we can use Lemmas 5.3, 5.4 to concatenate inner and boundary solutions.

Lemma 5.6 (Combination of Inner and Boundary Solution Intervals)
If an inner solution interval (t_1, t_2) follows a boundary solution interval, then we can specify λ_0 in (5.11) as $\lambda_0 = \dot{c}_{prod}(d(t_1)) - c_{inv}t_1$ and obtain

$$v(t) = \left(\dot{c}_{prod}(d(t))\right)^{-1} \left(\dot{c}_{prod}(d(t_1)) + c_{inv}(t - t_1)\right). \tag{5.16}$$

Moreover, the optimal production rate is continuous and positive for all $t \in [0, T]$

Given continuity and concatenatabiliy of the solution, we can conclude that once we are on a inner solution interval, we can either stay on it until the terminal time is reached, or continuously switch to a boundary solution interval. In particular, the following theorem holds:

Theorem 5.7 (Concatenation of Solution Intervals)
Suppose there exists an interval (σ_1, σ_2), where (5.13) does not hold, i.e.

$$c_{inv}(t) < \dot{d}(t)\ddot{c}_{prod}(d(t)) \quad \forall t \in (\sigma_1, \sigma_2) \tag{5.17}$$

Then, there exists an interval (t_1, t_2), which is an inner solution interval and satisfies $(\sigma_1, \sigma_2) \subset (t_1, t_2)$. The boundary points t_1, t_2 are given by

$$\int_{t_1}^{t_2} d(t)dt = \int_{t_1}^{t_2} \left(\dot{c}_{prod}(d(t))\right)^{-1} (\lambda_0 + c_{inv}t)\, dt \tag{5.18}$$

$$\lambda_0 = \dot{c}_{prod}\left(d(t_1)\right) - c_{inv}t_1 \quad \text{if } t_1 > 0 \tag{5.19}$$

$$\lambda_0 = \dot{c}_{prod}\left(d(t_2)\right) - c_{inv}t_2 \quad \text{if } t_2 < T. \tag{5.20}$$

From (5.18), we obtain that the areas under the curves of d and v are identical, i.e. the sum of demands equals the produced products. Equations (5.19), (5.20) state that if before or after the inner solution interval there is a boundary solution interval, then the production rate and the demand are identical at the beginning and at the end of the inner solution interval. Hence, the exit from and the entrance of an empty stock is tangential.

Note that the optimal production strategy is a smoothed version of the demand: An optimal production and inventory strategy has to weigh between the extremes of a smooth production with large fluctuations in the inventory and a production synchronous to demand without inventory. Hence, demand spikes are flattened and period of low demand are filled. The way the costs influence the production is given by its derivative

$$\dot{v}(t) = c_{\text{inv}} \frac{d}{dt}\left(\dot{c}_{\text{prod}}^{-1} v(t)\right) = \frac{c_{\text{inv}}}{\ddot{c}_{\text{prod}}(v(t))}$$

Remark 5.8
The flattening and filling is depending on the capacity of the inventory $\overline{z}$, i.e. only the maximal storage capacity can be used to smooth the optimal production rate.

Theorem 5.7 allows us to derive an optimal solution for any time interval. Additionally, we see that if $t^\star \in [0, T]$ is a time instant where $z(t^\star) = 0$, $v(t^\star) = d(t^\star)$ (boundary solution

interval), then the optimal solution in $[0, t^\star]$ is independent from changes in the rest interval $[t^\star, T]$ if the accumulated demand satisfies

$$\int_{t^\star}^{t} d(s)ds \leq \int_{t^\star}^{t} \dot{c}_{\text{prod}}^{-1}(\dot{c}_{\text{prod}}(d(t^\star)) + c_{\text{inv}}(s - t^\star))ds, \tag{5.21}$$

i.e. the inventory level is positive for all $t \in (t^\star, T]$.
For an inner solution interval, let $t_1 \in [0, T]$ be an instant with $z(t_1) > 0$ and let $t_2 \in (t_1, T]$ be the first time instant where $z(t_2) = 0$, i.e. the first instant after t_1 that the inventory is empty. If (5.21) holds, then we obtain the same independence, that is the solution on $[0, t_2]$ is independent from the solution on $(t_2, T]$.

This observation gives rise to the so called prediction and decision horizon.

5.4 Prediction and Decision Horizon

As we have seen in the previous section, for some dynamical optimal control problems it is not necessary to compute the optimal strategy for the entire planning interval immediately. It is more important to find the optimal solution for the next time steps with least possible information regarding the future development of the demand, of the costs and of the prices. To this end, we can utilize the independence property, which we have shown for the production and inventory problem. This property allows us to derive an optimal solution for a shorter optimization horizon, which is independent from the solution on the remaining part of the prediction horizon.

Here, we define these time instances as follows:

Definition 5.9 (Decision and Prediction Horizon)
Given an optimal control problem (OCP) from Definition 1.26 in the continuous version of Remark 1.27, where the planning horizon T may be infinite. If there exist time instances t_1, t_2 with $0 < t_1 \leq t_2 \leq T$ such that the optimal solution on $[0, t_1]$ is independent from the solution for $t \geq t_2$, then t_1 is called decision horizon, and t_2 is called prediction horizon.

Hence, to obtain the optimal solution on $[0, t_1] \subset [0, T]$, it is sufficient to look at the time interval $[0, t_2]$. This property is particularly important for the inventory problem. Here, we obtain:

Theorem 5.10
Suppose an optimal control problem (OCP) from Definition 1.26 in the continuous version of Remark 1.27 with constraints

$$\underline{x} \leq x(t) \leq \overline{x}$$

to be given. If there exists two instances $\tau_1, \tau_2 \in [0, T]$ such that the optimal solution satisfies $x(\tau_1) = \underline{x}$ and $x(\tau_2) = \overline{x}$, then $t_1 = \min(\tau_1, \tau_2)$ is the decision horizon and $t_2 = \max(\tau_1, \tau_2)$ is the prediction horizon.

Now the model functions, i.e. the demand, can change for any $t \geq t_2$ and even the terminal time T can be changed to any value $T \geq t_2$, the optimal solution on $[0, t_1]$ will remain unchanged.

5.5 Simultaneous Price and Production Decision

In the previous Section 5.1, we considered different variants of the Production and Inventory problem, where we supposed the demand to be fixed. Hence, by a given price development $p(t)$, the payoff $p(t)d(t)$ was not controllable. As a consequence, only the total costs had to be minimized.

Here, we assume the demand to be depending on the price $d(p(t), t)$, i.e. the price now is an additional degree of freedom within our optimal control problem. Hence, the payoff $p(t)d(p(t), t)$ can be controlled. To integrate this freedom in our production and inventory problem, we modify the cost function and utilize

$$J_T(p, v, z) = \int_0^T (p(t)d(p(t), t) - c_{\text{prod}}(v(t), t) - c_{\text{inv}}(z(t), t))\, dt + c_{\text{inv},T}(z(T), T). \tag{5.22}$$

The underlying dynamics is given by

$$\dot{z}(t) = v(t) - d(p(t), t), \quad z(0) = z_0, \tag{5.23}$$

and subject to the constraints

$$p(t) \geq 0, \quad v(t) \geq 0, \quad z(t) \geq 0 \quad \forall t \in [0, T], \tag{5.24}$$

which gives us the problem

$$\begin{aligned} &\text{maximize} \quad J_T(p, v, z) = \int_0^T (p(t)d(p(t), t) - c_{\text{prod}}(v(t), t) - c_{\text{inv}}(z(t), t))\, dt \\ &\qquad\qquad + c_{\text{inv},T}(z(T), T) \\ &\text{with respect to } p, v : [0, T] \to \mathbb{R}_0^+ \\ &\text{subject to } \dot{z}(t) = v(t) - d(p(t), t), \quad z(0) = z_0, \\ &\qquad 0 \leq v(t) \leq \overline{v},\ 0 \leq z(t) \leq \overline{z} \text{ and } p(t) \geq 0 \quad \forall t \in [0, T]. \end{aligned}$$

Different variants of this problem type only differ in the functional form of the demand d and the costs c_{prod}, c_{inv}, and possibly existing or non existing bounds on the production rate and the inventory stock.

5.6 Pekelman Model

Within the *Pekelman Model*, we assume that the demand is linearly and nonautonomously depending on the price $p(t)$ via

$$d(p(t), t) = \alpha(t) - \beta(t)p(t), \tag{5.25}$$

where $\alpha(t), \beta(t) > 0$ are given functions in time describing fluctuations. Moreover, we assume the production costs $c_{\text{prod}}(v(t), t)$ to be convex and the inventory costs to be linear, i.e. $c_{\text{inv}}(z(t), t) = c_{\text{inv}} z(t)$. Additionally, we impose the constraint

$$\dot{c}_{\text{prod}}(0) < \frac{\alpha(t)}{\beta(t)} \qquad \forall t \in [0, T]. \tag{5.26}$$

This condition ensures that the marginal costs of the first unit to be produced are larger or equal to the price α/β, for which demand is equal to zero. Otherwise, production is always

zero. Hence, the aim of a monopolist is to solve the problem

$$\begin{aligned}
&\text{maximize} \quad J_T(p,v,z) = \int_0^T (p(t)(\alpha(t)-\beta(t)p(t)) - c_{\text{prod}}(v(t)) - c_{\text{inv}}z(t))\,dt \\
&\qquad\qquad + c_{\text{inv},T}z(T) \\
&\text{with respect to } p, v : [0,T] \to \mathbb{R}_0^+ \\
&\text{subject to } \dot{z}(t) = v(t) - (\alpha(t) - \beta(t)p(t)), \quad z(0) = z_0, \\
&\qquad 0 \le v(t) \le \overline{v},\ 0 \le z(t) \le \overline{z} \text{ and } 0 \le p(t) \le \tfrac{\alpha(t)}{\beta(t)} \quad \forall t \in [0,T].
\end{aligned}$$

Now, we can proceed as for the Arrow–Karlin model. Recall that the appearance of an inner solution interval can be seen from condition (5.17). The condition states that the demand increases minimally at a rate, which is proportional to the inventory costs and indirect proportional to $\ddot{c}_{\text{prod}}$. In the present Pekelman model, the demand d depends on the price p, and we obtain that an exogenous function $\varphi(t)$ takes the role of $d(t)$.

Lemma 5.11 (Marginal Revenue)
The equation

$$\dot{c}_{prod}^{-1}(\varphi(t)) = \frac{1}{2}\left(\alpha(t) - \beta(t)\varphi(t)\right) \tag{5.27}$$

exhibits a unique solution $\varphi(t)$ for each $t \in [0,T]$. This solution is continuously differentiable and satisfies

$$\dot{c}_{prod}(0) < \varphi(t) < \frac{\alpha(t)}{\beta(t)}. \tag{5.28}$$

The function φ can be interpreted as marginal revenue of the static problem without inventory $\max_v(p(t)v(t) - c_{\text{prod}}(v(t)))$ with $v(t) = \alpha - \beta p(t)$. Eliminating p, first order necessary conditions reveals

$$\dot{c}_{\text{prod}}(v(t)) = \frac{d}{dv}\left(p(v(t))v(t)\right) = \frac{\alpha(t)}{\beta(t)} - \frac{2v(t)}{\beta(t)}.$$

Hence, (5.27) holds true for $\varphi(t) := \frac{\alpha(t)}{\beta(t)} - \frac{2v(t)}{\beta(t)}$.

Utilizing the function $\varphi(t)$, we can continue similar to the Arrow–Karlin model and obtain:

Lemma 5.12 (Boundary Solution Interval)
On a boundary solution interval $z(t) = 0$ the conditions

$$\lambda(t) = \varphi(t) \tag{5.29}$$

$$v(t) = \dot{c}_{prod}^{-1}(\varphi(t)) > 0 \tag{5.30}$$

$$0 < p(t) = \frac{1}{2}\left(\frac{\alpha(t)}{\beta(t)} + \varphi(t)\right) < \frac{\alpha(t)}{\beta(t)} \tag{5.31}$$

hold and

$$c_{inv} \ge \dot{\varphi}(t). \tag{5.32}$$

As a consequence, if the inventory costs c_{inv} are sufficiently large, i.e. $c_{\text{inv}} \geq \max \dot{\varphi}(t)$, then we have $\varphi = \lambda$ due to (5.29), i.e. $\varphi(t)$ represents the value of the first element in the inventory at all times. Moreover, we can conclude

Theorem 5.13 (Full Boundary Solution)
If $z_0 = 0$, $c_{inv,T} < \varphi(T)$ and (5.32) holds for all $t \in [0,T]$, then the boundary solution (5.30) is optimal on $[0,T]$.

Similarly, if the initial inventory level is larger than the total demand, then we obtain a full inner solution.

Theorem 5.14 (Full Inner Solution)
If $c_{inv,T} = 0$ and

$$z_0 > \int_0^T \min\{\alpha(t), \frac{1}{2}(\alpha(t) - c_{inv}\beta(t)(t-T))\}dt, \tag{5.33}$$

then the initial inventory will not be consumed and no products will be produced. Particularly, we have

$$z(t) > 0 \tag{5.34}$$
$$v(t) = 0 \tag{5.35}$$
$$\lambda(t) = c_{inv}(t-T) \tag{5.36}$$
$$p(t) = \max\left\{0, \frac{1}{2}\left(\frac{\alpha(t)}{\beta(t)} + c_{inv}(t-T)\right)\right\} \tag{5.37}$$

for all $t \in [0,T]$.

Similar to boundary solution intervals, we can also characterize inner solution intervals.

Lemma 5.15 (Inner Solution Interval)
If there exists an interval (σ_1, σ_2), where (5.32) does not hold, then there exists an interval $(t_1, t_2) \supset (\sigma_1, \sigma_2)$ with an inner solution satisfying

$$\lambda(t) = \lambda_0 + c_{inv}t \tag{5.38}$$
$$\dot{z}(t) \begin{cases} > & \text{if } \lambda(t) > \varphi(t) \\ = & \text{if } \lambda(t) = \varphi(t) \\ < & \text{if } \lambda(t) < \varphi(t) \end{cases}. \tag{5.39}$$

- *For $t_1 > 0$, $t_2 < T$, the parameter λ_0, t_1, t_2 are given by*

$$\int_{t_1}^{t_2} v(\lambda(t)) - \alpha(t) + \beta(t)p(\lambda(t))dt = 0 \tag{5.40}$$
$$\lambda(t_1) = \varphi(t_1) \tag{5.41}$$
$$\lambda(t_2) = \varphi(t_2) \tag{5.42}$$

with

$$v(\lambda(t)) = \begin{cases} 0 & \text{if } \lambda(t) \leq 0 \\ \dot{c}_{prod}^{-1}(\lambda(t)) & \text{if } \lambda(t) < 0, \end{cases} \tag{5.43}$$

$$p(\lambda(t)) = \begin{cases} 0 & \text{if } \lambda(t) \leq \alpha(t)/\beta(t) \\ \frac{1}{2}(\alpha(t)/\beta(t) + \lambda(t) & \text{if } -\alpha(t)/\beta(t) < \lambda(t) < \alpha(t)/\beta(t) \\ \alpha(t)/\beta(t) & \text{if } \lambda(t) \geq \alpha(t)/\beta(t). \end{cases} \tag{5.44}$$

- *If* $t_1 = 0$*, then* (5.40) *is replaced by*

$$z_0 + \int_0^{t_2} v(\lambda(t)) - \alpha(t) + \beta(t)p(\lambda(t))dt = 0 \tag{5.45}$$

  and (5.41) *can be dropped.*

- *If* $t_2 = T$*, then* (5.42) *can be dropped. If the resulting* $\lambda(T) < c_{inv,T}$*, then* (5.40) *is replaced by*

$$\lambda(T) = c_{inv,T} \tag{5.46}$$

  and we have $z(T) > 0$*. Else,* (5.40) *holds and we have* $z(T) = 0$*.*

If $z_0 > 0$ *or* $c_{inv,T} > \varphi(T)$ *holds, then an inner solution interval has to be chosen at the beginning or the end respectively, independent of* (5.38)*.*

Combined, we see that for both the Arrow–Karlin and the Pekelman model, the optimal solution can be concatenated from boundary and inner pieces. The structure of these pieces always follows the same principles, which implicitly arise from Pontryagin's Maximum Principle, cf. [2, Chapter 1], which is also called the *indirect approach*. This insight into solution properties allows us to check whether a direct approach — first discretize the optimal control problem (OCP), then solve the resulting optimization problem — reveals a good solution. Note that even in the indirect case, we still need to numerically evaluate the solution.

Chapter 6

Maintenance and Replacement

In the previous chapter, we considered the problem of optimal usage of production capacities in terms of profit and connected costs for production and inventory. To this end, machines used to produce the respective goods, which are subject to wearout. Within the present chapter, we focus on the optimal planning of wear reduction and regenerative activities, i.e. we want to simultaneously compute the optimal restoration intensity of a machine and the respective optimal point of replacement.

Production facilities are wearing out over time proportionally to their workload and/or may suddenly fail. Hence, any machine is subject to (deterministic) wearout and thereby loss in value, which is also affected by technological improvements, but also subject to (stochastic) risk of a sudden breakdown. Here, we discuss some fundamental control theoretic maintenance models. These models contain only one state variable, that is the reliability or the resale value of a machine, and the two controls preventive maintenance investments and intensity of use of the machine. For these models, prominent characteristics are the free terminal time denoting the point of replacement, which is a third control value, and the time dependency of model parameters.

Within this chapter, we first formalize the problem setting before we present two different model types. Within the Kamien–Schwartz Model we aim to reduce the risk of a sudden machine breakdown by taking preventive actions. In the second model, the Thompson model, we incorporate a deterministic wearout of a machine, which we try to reduce to generate an optimal solution.

6.1 Problem Formulation

The maintenance problem is a non autonomous control problem with state x denoting the condition of the machine, and two control u and v representing the maintenance and usage rate respectively. The problem then reads

$$
\begin{aligned}
&\text{maximize} \quad J_T(u,v) = \int_0^T \exp^{-rt} c(x,u,v,t)dt + \exp^{-rT} c_{u,T}(x(T),T) \\
&\text{with respect to } u, v : [0,T] \to \mathbb{R}_0^+ \\
&\text{subject to } \dot{x}(t) = f(x(t),u(t),v(t),t), \quad x(0) = x_0.
\end{aligned}
$$

Here, we particularly assume the following to hold:

Assumption 6.1
For the maintenance and replacement problem, the following properties shall hold:

$$\begin{aligned}
&\frac{\partial f}{\partial x} < 0, \quad \frac{\partial c}{\partial x} > 0, \quad \frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 c}{\partial x^2} = 0, \quad \frac{\partial c_{u,T}}{\partial x} \geq 0 \\
&\frac{\partial f}{\partial u} > 0, \quad \frac{\partial c}{\partial u} < 0, \quad \frac{\partial^2 f}{\partial x \partial u} \geq 0 \\
&\frac{\partial f}{\partial v} < 0, \quad \frac{\partial c}{\partial v} > 0, \quad \frac{\partial^2 f}{\partial x \partial v} \leq 0 \\
&\frac{\partial^2 f}{\partial x \partial t} \leq 0, \quad \frac{\partial^2 c}{\partial x \partial t} \leq 0
\end{aligned}$$

We can interpret these assumptions as follows:

- Investments u induce costs, i.e. $\frac{\partial c}{\partial u} < 0$, but improve the state of the machine, i.e. $\frac{\partial f}{\partial u} > 0$.
- The usage rate v induces profits, i.e. $\frac{\partial c}{\partial v} > 0$, but has a negative effect on the machine, i.e. $\frac{\partial f}{\partial v} < 0$.

The following results also hold for general control systems. Therefore, we utilize the notation from Chapter 1, where (OCP) is given in Definition 1.26. For such problems, the notion of state separability can be introduced.

Definition 6.2 (State Separability)
A control problem is called state separable if the Hamiltonian

$$H(x, u, \lambda, t) := c(x, u, t) + \lambda f(x, u, t)$$

with state x, control u and adjoint λ satisfies

$$\frac{\partial^2 H}{\partial x^2} = 0, \qquad \frac{\partial^2 H}{\partial x \partial u} = 0 \text{ for } \frac{\partial H}{\partial x} = 0, \qquad \text{and } \frac{\partial^2 c_{u,T}}{\partial x^2} = 0.$$

If a control system is state separable, one can show that the adjoint is monotone, and that all control variables inherit this property. To this end, we introduce the equilibrium of the adjoint

$$\hat{\lambda}(t) = \frac{\frac{\partial c}{\partial x}(x, u, t)}{\left(r - \frac{\partial f}{\partial x}(x, u)\right)}. \tag{6.1}$$

As we would expect for the maintenance problem, the shadow price λ is decreasing over time representing the sales price of the machine. Moreover, this decrease induces decreasing maintenance costs u and an increasing usage rate v.

Lemma 6.3 (Monotonicity of Solutions)
Suppose Assumptions 6.1 hold for a state separable maintenance and replacement problem with $\hat{\lambda}(t) > 0$ *and*

$$\dot{\lambda}(t) \begin{cases} > 0 \\ = 0 \\ < 0 \end{cases} \iff \lambda(t) \begin{cases} > \hat{\lambda}(t) \\ = \hat{\lambda}(t) \\ < \hat{\lambda}(t) \end{cases} . \tag{6.2}$$

Moreover, if $\dot{\lambda}(\bar{t}) \geq 0$ *for some* $\bar{t} \in [0, T]$*, then we have for all* $t \in [0, T]$

$$\dot{\hat{\lambda}}(\bar{t}) \leq 0, \quad \dot{\lambda}(t) \geq 0, \quad \dot{u}(t) \geq 0, \quad \dot{v}(t) \leq 0. \tag{6.3}$$

Corollary 6.4
Given the assumptions of Lemma 6.3 hold and $\dot{\lambda}(T) < 0$*, then we have* $\dot{\lambda}(t) < 0$*,* $\dot{u}(t) < 0$ *and* $\dot{v}(t) > 0$ *for all* $t \in [0, T]$*.*

Additionally to monotonicity properties of the adjoint (or shadow price), we can also show properties of the terminal time. Recall that in the maintenance and replacement problem, the terminal time corresponds to the time of replacement.

Lemma 6.5 (Replacement)
Suppose $T^\star > 0$ *is the optimal terminal time for the maintenance and replacement problem and assumptions from Lemma 6.3 hold. If the elasticities satisfy*

$$\sigma_{f,x} = \frac{\frac{\partial f}{\partial x}(x, u, v, T) \cdot x}{f(x, u, v, T)} \geq 1 \tag{6.4}$$

$$\sigma_{c,x} = \frac{\frac{\partial c}{\partial x}(x, u, v, T) \cdot x}{c(x, u, v, T)} \geq 1 \tag{6.5}$$

$$\sigma_{c_{u,T},x} = \frac{\frac{\partial c_{u,T}}{\partial x}(x, T) \cdot x}{c_{u,T}(x, T)} \leq 1 \tag{6.6}$$

and

$$\frac{\partial c_{u,T}}{\partial T}(x, T) \leq 0 \tag{6.7}$$

for each x*,* u*,* v *and* $T = T^\star$*, then we have*

$$\dot{\lambda}(T^\star) \leq 0. \tag{6.8}$$

Hence, we can conclude

Lemma 6.6 (Strict Monotonicity)
Suppose a state separable maintenance and replacement problem to be given and Assumption 6.1

to hold. Moreover, the elasticities satisfy (6.4), (6.5), (6.6) *and inequality* (6.7) *holds. Then we have*

$$\dot{\lambda}(t) < 0, \quad \dot{u}(t) < 0, \quad \textit{and} \quad \dot{v}(t) > 0 \qquad \forall t \in [0, T]. \tag{6.9}$$

Now, Lemmata 6.5 and 6.6 allow us to draw conclusions regarding monotonicity of the optimal maintenance and replacement control.

6.2 Kamien–Schwartz Model

Within this model, a sudden machine breakdown may occur stochastically. Once such an event has taken place, the machine cannot be repaired, yet preventive actions may be taken to extend the lifespan of the machine denoted by Ξ. The probability density function of the lifespan shall be given by $\mathrm{P}(\Xi \leq t)$, which allows us to formulate the natural failure rate via

$$h(t) = \lim_{\Delta \to 0} \frac{1}{\Delta} \mathrm{P}\left(t < \Xi \leq t + \Delta \mid \Xi > t\right) = \frac{\dot{\mathrm{P}}(\Xi \leq t)}{1 - \mathrm{P}\left(\Xi \leq t\right)} \tag{6.10}$$

To control the process, the maintenance rate u can be used. Here, $100u$ corresponds to the percentage at which the failure rate is decreased. From (6.10), we directly obtain

$$\dot{\mathrm{P}}(\Xi \leq t) = \left(1 - u(t)\right) h(t) \left(1 - \mathrm{P}\left(\Xi \leq t\right)\right), \tag{6.11}$$

which coincides with (6.10) for $u = 0$ while for $u = 1$ the failure rate and density function $\dot{\mathrm{P}}$ vanishes.

Within the Kamien–Schwartz Model, the reliability $x(t) = 1 - \mathrm{P}(\Xi \leq t)$ is used as the state of the system. This reveals the dynamics

$$\dot{x}(t) = -\left(1 - u(t)\right) h(t) x(t), \quad x(0) = 1. \tag{6.12}$$

Moreover, we denote the costs for maintenance by $c_u(u)$, the profit by operating a machine per time unit by c_v, the value of an operational machine at time t by $V(t)$ and the value of a broken machine by W. For these variables, we consider the following assumption:

Assumption 6.7
The natural failure rate is (weakly) monotone increasing, i.e.

$$h(t) \geq 0, \quad \dot{h}(t) \geq 0, \tag{6.13}$$

and that the maintenance rate is bounded by

$$0 \leq u(t) \leq 1 \qquad \forall t \in [0, T]. \tag{6.14}$$

Moreover, the machine shall not fail instantly, i.e. $\mathrm{P}(\Xi \leq 0) = 0$. The maintenance costs to reduce the failure rate is over-proportionally increasing, that is

$$c_u(0) = 0, \quad \dot{c_u}(u) > 0, \quad \ddot{c_u}(u) > 0 \qquad \text{for } u \in (0, 1). \tag{6.15}$$

For simplicity of exposition, we additionally assume that the costs for a small reduction of the failure rate is almost zero and if failure is to be neglected, then the corresponding costs are infinite, i.e.

$$\dot{c_u}(0) = 0, \quad \dot{c_u}(1) = \infty. \tag{6.16}$$

Last, c_v and W are positive constants and the resale value $V(t)$ is monotone decreasing with

$$\dot{V}(t) \leq 0, \quad 0 \leq W \leq V(t) \leq c_v/r. \tag{6.17}$$

Note that these assumptions are economically meaningful: the resale value of an operational machine is never increasing and always higher than the scrap value. It is, however, smaller than the operating revenue of the machine.

Remark 6.8
While unimportant for the structure of the optimal control, condition (6.16) *rules out the boundary solutions* $u = 0$ *and* $u = 1$.

Since the gain and the maintenance costs only arise for an intact machine, the expected discounted net gains of operating and selling an intact machine is given by

$$\int_0^T \exp^{-rt} \left(c_v - c_u(u(t))h(t)\right) x(t)dt + \exp^{-rT} V(T)x(T), \tag{6.18}$$

and the expected net gain of scrapping a broken machine in the case $\Xi \leq T$ reads

$$\int_0^T \exp^{-rt} W\dot{\mathrm{P}}(\Xi \leq t)dt = Wx_0 - \exp^{-rT} Wx(T) - rW \int_0^T \exp^{-rt} x(t)dt. \tag{6.19}$$

These two values can now be combined to form a cost functional. Since Wx_0 is constant, we can neglect it and obtain the optimal control problem

$$\begin{aligned}
&\text{maximize} \quad J_T(u,v) = \int_0^T \exp^{-rt} \left(c_v - rW - c_u(u(t))h(t)\right) x(t)dt \\
&\qquad\qquad + \exp^{-rT} \left(V(T) - W\right) x(T) \\
&\text{with respect to } u, v : [0,T] \to \mathbb{R}_0^+ \\
&\text{subject to } \dot{x}(t) = -\left(1 - u(t)\right) h(t)x(t), \quad x(0) = 1 \\
&\qquad 0 \leq u(t) \leq 1 \qquad \forall t \in [0,T].
\end{aligned}$$

Hence, we obtain the maintenance and replacement problem from Section 6.1 with

$$\begin{aligned}
c(x,u,v,t) &:= c_v - rW - c_u(u(t))h(t) \\
c_{u,T}(x(T),T) &:= \left(V(T) - W\right) x(T) \\
f(x(t),u(t),v(t),t) &:= -\left(1 - u(t)\right) h(t)x(t).
\end{aligned}$$

Since Assumption 6.7 induces Assumption 6.1, and since the term $(c_u - W)u$ is linear in u, the model is state separable. Hence, by Lemma 6.3 it follows that

Theorem 6.9 (Solution Properties)
Given the Kamien–Schwartz Model and Assumption 6.7 holds, then the conclusion

$$\dot{\lambda}(\bar{t}) \geq 0 \quad \Longrightarrow \quad \dot{\lambda}(t) \geq 0,\ \dot{u} \geq 0$$

holds for all $t \geq \bar{t}$. Additionally, since by optimality $\dot{\lambda}(t) = \ddot{c_u}(u(t))\dot{u}(t)$ and by assumption $\ddot{c_u}(u(t) > 0$, we have

$$\operatorname{sgn}(\dot{u}(t)) = \operatorname{sgn}(\dot{\lambda}(t)).$$

To apply Lemma 6.5, we observe

$$\sigma_{f,x} = \sigma_{c_u,x} = \sigma_{c_{u,T},x} = 1$$
$$\frac{\partial c_{u,T}}{\partial T}(x,T) = x(T)\dot{V}(T).$$

Hence, the following result holds:

Theorem 6.10 (Solution Properties)
Given the Kamien–Schwartz Model and Assumption 6.7 holds, then for optimal replacement at time $T^\star$ the optimal maintenance strategy $u(t)$ satisfies the monotonicity condition

$$\dot{u}(t) \begin{cases} < 0 & \text{if } \dot{V}(T^\star) < 0 \\ \leq 0 & \text{if } \dot{V}(T^\star) = 0 \end{cases}.$$

Therefore, if the time of replacement $T^\star$ is chosen optimally, then the reduction of the failure rate is decreasing. Since the natural failure rate $h(t)$ is increasing, the true failure rate $(1 - u(t))\, h(t)$ is increasing as well. Whether or not the costs of a preventive maintenance $c_u(u(t))h(t)$ are de- or increasing, fully depends on the case itself.

6.3 Thompson Model

The Kamien–Schwartz model aims at reducing the risk of a sudden machine breakdown by taking preventive actions. Within the Thompson model, this aim is changed to reduce deterministic wearout. Again, we define T as the time to replace the machine and denote the age of a machine by t. The value of a new machine is represented by $y(0) = y_0$ and $u(t)$ are the maintenance actions taken in period t. The effectiveness of a maintenance action is given by $g(t)$, which states by how much the reduction of the resale price is reduced if we invest in maintenance. The loss in value of a machine consists of two components:

- The technical obsolescence $\gamma(t)$ reflects the loss in value due to new inventions, availability of more powerful machines and the decaying productivity and resale value due to age.
- The wear rate $\delta(t)$ describes the loss in value due to deterministic wearout.

Hence, we obtain the dynamic of the resale price of a machine based on obsolescence, wear rate and maintenance via

$$\dot{y}(t) = -\gamma(t) - \delta(t)y(t) + g(t)u(t). \tag{6.20}$$

For the Thompson model, we consider the following set of assumptions to hold:

Assumption 6.11
The effectiveness of the maintenance $g(t)$ is monotone decreasing with the age of the machine, while obsolescence $\gamma(t)$ and wear rate $\delta(t)$ are monotone increasing, i.e.

$$\dot{g}(t) \geq 0, \quad \dot{\gamma}(t) \leq 0, \quad \dot{\delta}(t) \leq 0. \tag{6.21}$$

Similar to the Kamien–Schwartz model, the maintenance rate is bounded by

$$0 \leq u(t) \leq \overline{u} \qquad \forall t \in [0, T]. \tag{6.22}$$

Moreover, the profit of a machine with value $y(t)$ is given by $c_{\text{prod}}(t)y(t)$ and the costs for maintenance are given by $c_u u(t)$.

Combining the profit of a machine with its maintenance costs and a discount factor and selling the machine at the terminal time instant T, then the current value of the monetary flow is given by

$$J_T(u, T) = \int_0^T \exp^{-rt} \left(c_{\text{prod}}(t)y(t) - c_u u(t)\right) dt + \exp^{-rT} y(T). \tag{6.23}$$

The combined optimal control problem forms the Thompson model

$$\begin{aligned}
&\text{maximize} \quad J_T(u, T) = \textstyle\int_0^T \exp^{-rt} \left(c_{\text{prod}}(t)y(t) - c_u u(t)\right) dt \\
&\qquad\qquad\qquad\qquad + \exp^{-rT} y(T) \\
&\text{with respect to } u : [0, T] \to \mathbb{R}_0^+ \\
&\text{subject to } \dot{y}(t) = -\gamma(t) - \delta(t)y(t) + g(t)u(t), \quad y(0) = y_0 \\
&\qquad\qquad 0 \leq u(t) \leq \overline{u} \qquad \forall t \in [0, T].
\end{aligned}$$

For the Thompson model, the shadow price is given by

$$\dot{\lambda}(t) = (r + \delta(t))\, \lambda(t) - c_{\text{prod}}(t), \qquad \lambda(T) = 1 \tag{6.24}$$

and independent from $y(t)$ and $u(t)$ and therefore can be solved independently. Here, $\lambda(t)$ represents the cost/profit of an additional unit of the machine at time t. The shadow price is split into the part measuring the increase of the resale value at time T, and the part measuring the increase of the production profit from t to T if the resale value at time t is increased. To obtain a meaningful case, we impose the following:

Assumption 6.12
For the Thompson model we suppose that

$$c_{\text{prod}}(t) > r + \delta(t) \quad \forall t \in [0, T] \quad \text{and } c_{\text{prod}}(T) > r + \delta(T). \tag{6.25}$$

Following Assumption 6.12, at each time instant t we gain more profit from one unit of machine value x than we loose by discounting and wearout, i.e. running the machine pays off.

Similar to the Kamien–Schwartz model, we can now conclude monotonicity of the shadow prices using Lemma 6.3:

Theorem 6.13 (Monotonicity of Solutions)
Given the Thompson Model, suppose Assumptions 6.11 and 6.12 to hold. Then we have

$$\dot{\lambda}(T) < 0. \tag{6.26}$$

Additionally, the function $\hat{\lambda}(t) = c_{prod}(t)/\left(r + \delta(t)\right)$ *is monotone decreasing and*

$$\dot{\lambda}(t) = (r + \delta(t)) \cdot \left(\lambda(t) - \hat{\lambda}(t)\right) \tag{6.27}$$

$$\dot{\lambda}(t) < 0 \tag{6.28}$$

holds for all $t \in [0, T]$.

Here, due to linearity of the control $u(t)$ and the separability of the shadow price $\lambda(t)$ from the value of the machine $y(t)$, one can show the following:

Theorem 6.14 (Solution Properties)
If Assumptions 6.11 and 6.12 hold for the Thompson model, then

$$u(t) = \begin{cases} 0 & \text{if } \lambda(t)g(t) < 1 \\ \text{not defined} & \text{if } \lambda(t)g(t) = 1 \\ \overline{u} & \text{if } \lambda(t)g(t) > 1 \end{cases} \tag{6.29}$$

is the optimal maintenance strategy.

Unfortunately, Theorem 6.14 reveals only the optimal boundary controls and does not state what should be done if the marginal revenue $\lambda(t)g(t)$ of a maintenance unit $u(t)$ equals a maintenance unit, i.e. $\lambda(t)g(t) = 1$. Basically, (6.29) states that if the payoff of one maintenance unit is larger than the price of the maintenance unit, then the machine is maintained, otherwise it is not maintained.

Since $g(t)$ is monotone decreasing and $\lambda(t)$ is strictly monotone decreasing according to (6.28), then also $\lambda(t)g(t)$ exhibits this property. Hence, no inner solution can occur and we can show:

Corollary 6.15 (Solution)
For a Thompson model satisfying Assumptions 6.11 and 6.12, the optimal maintenance strategy is given by

$$u(t) = \begin{cases} 0 & \text{for } 0 \le t \le \tau \\ \overline{u} & \text{for } \tau \le t \le T \end{cases} \tag{6.30}$$

for some $\tau \in [0, T]$ *satisfying* $\lambda(\tau)g(\tau) = 1$. *If the latter equation reveals* $\tau < 0$ *we set* $\tau = 0$, *and if it reveals* $\tau > T$ *we set* $\tau = T$.
Additionally, the optimal replacement time T *is given as the unique solution of*

$$y(T) = \frac{\gamma(T)}{c_{prod}(T) - r - \delta(T)}. \tag{6.31}$$

Appendices

Appendix A

Theoretical Results

A.1 Unconstrained Optimization

The basis of the analysis of nonlinear optimization problems is given by Taylor's Theorem:

Theorem A.1 (Taylor's Theorem)
Consider a function $F : \mathbb{R}^{n_z} \to \mathbb{R}$ which is continuously differentiable and a direction vector $d \in \mathbb{R}^{n_z}$. Then we have

$$F(z+d) = F(z) + \nabla F(z+td)^\top d \tag{A.1}$$

for some $t \in (0,1)$. If F is twice continuously differentiable, then we also have

$$F(z+d) = F(z) + \nabla F(z)^\top d + \frac{1}{2} d^\top \nabla^2 F(z+td) d \tag{A.2}$$

for some $t \in (0,1)$.

Proof. Using the fundamental theorem of calculus, we have

$$F(z+d) = F(z) + \int_0^1 \frac{d}{dt} F(z+td) dt.$$

By the mean value theorem, there exist a $t \in (0,1)$ with

$$\int_0^1 \frac{d}{dt} F(z+td) dt = \frac{d}{dt} F(z+td) = \nabla F(z+td)^\top d,$$

where we used the chain rule for the second equality. This shows(A.1). By partial integration we further obtain

$$\int_0^1 \frac{d}{dt} F(z+td) dt = \frac{d}{dt}\Big|_{t=0} F(z+td) + \int_0^1 (1-t) \frac{d^2}{dt^2} F(z+td) dt$$

and again using the mean value theorem we get

$$\int_0^1 (1-t) \frac{d^2}{dt^2} F(z+td) dt = \frac{d^2}{dt^2} F(z+t'd) \int_0^1 (1-t) dt = \frac{1}{2} \frac{d^2}{dt^2} F(z+t'd)$$

for some $t' \in (0,t)$. Since by the chain rule we have

$$\left.\frac{d}{dt}\right|_{t=0} F(z+td) = \nabla F(z)^\top d \quad \text{and} \quad \frac{1}{2}\frac{d^2}{dt^2}F(z+t'd) = \frac{1}{2}d^\top \nabla^2 F(z+t'd)d$$

this shows (A.2). □

The advantage of Taylor's theorem is that it allows us to introduce knowledge on the gradient $\nabla F(z^\star)$ and the Hessian $\nabla^2 F(z^\star)$ into the search for a local minimizer $z^\star$. In particular, first order necessary conditions are derived very easily.

Theorem A.2 (First Order Necessary Conditions)
Consider a vector $z^\star \in \mathbb{R}^{n_z}$ and a function $F : \mathbb{R}^{n_z} \to \mathbb{R}$ where F is continuously differentiable in an open neighborhood of $z^\star$ and $z^\star \in \mathbb{R}^{n_z}$ is a local minimizer of F. Then we have $\nabla F(z^\star) = 0$.

Proof. Suppose $\nabla F(z^\star) \neq 0$ and set $d := -\nabla F(z^\star)$. Then we get $d^\top \nabla F(z^\star) = -\|\nabla F(z^\star)\|^2 < 0$. Since ∇F is continuous in a neighborhood of $z^\star$, there exists a scalar $T > 0$ such that $d^\top \nabla F(z^\star + td) < 0$ holds for all $t \in [0,T]$. By (A.1), for any $\bar{t} \in (0,T]$ we have $F(z^\star + \bar{t}d) = F(z^\star) + \bar{t}d^\top \nabla F(z^\star + td)$ for some $t \in (0,\bar{t})$. This implies $F(z^\star + \bar{t}d) < F(z^\star)$ for all $\bar{t} \in (0,T]$ which contradicts the local minimizer property of $z^\star$. □

In a similar manner, information on the Hessian can be used to derive second order necessary conditions from equation (A.2).

Theorem A.3 (Second Order Necessary Conditions)
Consider a vector $z^\star \in \mathbb{R}^{n_z}$ and a function $F : \mathbb{R}^{n_z} \to \mathbb{R}$ where F is twice continuously differentiable in an open neighborhood of $z^\star$ and $z^\star \in \mathbb{R}^{n_z}$ is a local minimizer of F. Then we have $\nabla F(z^\star) = 0$ and the Hessian $\nabla^2 F(z^\star)$ is positive semidefinite.

Proof. From Theorem A.2 we know that $\nabla F(z^\star) = 0$. Now, suppose $\nabla^2 F(z^\star)$ is not positive semidefinite and choose a vector d such that $d^\top \nabla^2 F(z^\star) d < 0$ holds. Using continuity of $\nabla^2 F(z^\star)$ in a neighborhood of $z^\star$, we know that there exists a scalar $T > 0$ such that $d^\top \nabla^2 F(z^\star + td)d < 0$ holds for all $t \in [0,T]$. Hence, using (A.2), for any $\bar{t} \in (0,T]$ and some $t \in (0,\bar{t})$ we obtain

$$F(z^\star + \bar{t}d) = F(z^\star) + \bar{t}\nabla F(z^\star)^\top d + \frac{1}{2}\bar{t}d^\top \nabla^2 F(z^\star + td)d\bar{t} < F(z^\star).$$

Similar to the proof of Theorem A.2, F is strictly decreasing along the direction d which contradicts the local minimizer property of $z^\star$. □

The results from Theorems A.2 and A.3 reveal guidelines to what we are looking for, i.e., which properties a local minimizer must fulfill. However, these results cannot be used to identify a local minimizer once we have found a candidate satisfying the previous conditions. In order to perform such a check, the following theorem can be used.

Theorem A.4 (Second Order Sufficient Conditions)
Consider a vector $z^\star \in \mathbb{R}^{n_z}$ and a function $F : \mathbb{R}^{n_z} \to \mathbb{R}$ where F is twice continuously differentiable in an open neighborhood of $z^\star$. If $\nabla F(z^\star) = 0$ and $\nabla^2 F(z^\star)$ is positive definite, then $z^\star$ is a local minimizer of F.

Proof. Due to F being twice continuously differentiable there exists a radius $r > 0$ such that $\nabla^2 F(z)$ is positive definite for all $z \in \{z \mid \|z - z^\star\| < r\}$. Now take any vector $d \in \mathbb{R}^{n_z}$ with $\|d\| < r$, then we have $z^\star + d \in \{z \mid \|z - z^\star\| < r\}$ and

$$\begin{aligned} F(z^\star + d) &= F(z^\star) + d^\top \nabla F(z^\star) + \frac{1}{2} d^\top \nabla^2 F(z^\star + td) d \\ &= F(z^\star) + \frac{1}{2} d^\top \nabla^2 F(z^\star + td) d \end{aligned}$$

for some $t \in (0, 1)$. Since $(z^\star + td) \in \{z \mid \|z - z^\star\| < r\}$, we have $d^\top \nabla^2 F(z^\star + td) d > 0$ and therefore $F(z^\star + d) > F(z^\star)$ holds showing the assertion. □

Bibliography

[1] W. Alt. *Nichtlineare Optimierung: Eine Einführung in Theorie, Verfahren und Anwendungen*. Springer, 2013.

[2] G. Feichtinger and R. Hartl. *Optimale Kontrolle ökonomischer Prozesse*. de Gryuter, 1986.

[3] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[4] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer, 2002.

[5] M. Gerdts. Optimierung. Technical report, Universität der Bundeswehr München, München, 2015.

[6] H. Khalil. *Nonlinear Systems*. Prentice Hall PTR, 2002.

[7] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. 2002.

[8] K. Neumann and M. Morlock. *Operations Research*. 2002.

[9] J. Nocedal and S. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[10] E. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, 1998.