





## **Eigenständigkeitserklärung**

Ich versichere, dass ich meine Masterarbeit eigenständig verfasst habe. Für die Verfassung dieser Arbeit habe ich jenes Wissen, das ich in den zahlreichen Lehrveranstaltungen meines Studiums sammeln durfte, sowie die am Ende dieser Arbeit angeführte Literatur verwendet.

Daniel Obszelka, Bakk.



## Vorwort

„Und worum geht es in deiner Arbeit?“, wurde ich immer wieder gefragt.

Es geht um Projektportfoliooptimierung unter einem Kompetenzentwicklungsmodell. Uns stehen verschiedene Projekte zur Verfügung, die wir bearbeiten können. Ziel ist es, die verfügbare Arbeitszeit derart auf diese Projekte aufzuteilen, dass der Gesamtertrag maximal ist.

Dabei sind für die Erledigung der Projekte verschiedene Kompetenzen in unterschiedlichem Ausmaß erforderlich. Machen wir von einer Kompetenz oft Gebrauch, werden wir besser in ihr und können in weiterer Folge Projekte, welche diese Kompetenzen benötigen, schneller bearbeiten. Das gleiche gilt aber auch umgekehrt.

Die Quelle der Entstehung dieser Arbeit ist in [Gut11] zu finden. Mich hat das Thema sofort gepackt! Anfangs war eigentlich an eine Simulationsstudie mit unsicheren Projekterträgen gedacht. Nachdem ich aber schon bei „einfachen“ Optimierungsroutinen mit einigen Problemen zu kämpfen hatte, begann ich ins Detail zu gehen. Und als ich damit begonnen hatte, die Langfristigkeit zu studieren, packte mich der Ehrgeiz!

Diese Arbeit ist viel mathematischer geworden als anfangs gedacht. Dennoch wollte ich sie für eine möglichst große Zielgruppe verständlich gestalten...

Mir war von Beginn an klar, dass ich ein Beispiel finden muss, mit dem sich jeder identifizieren kann und das mitreißt! Das einfach ist, damit die Ergebnisse und Ideen grafisch darstellbar sind. Aber komplex genug, um möglichst viele Theorien mit diesem Beispiel veranschaulichen zu können.

Dann kamen die Olympischen Sommerspiele 2012 in London. Österreich hatte bis kurz vor Schluss keine einzige Medaille erringen können. Die Favoriten waren der Reihe nach gescheitert und immer wieder wurden neue „Favoriten mit Außenseiterchancen“ aus dem Hut gezaubert.

Plötzlich war die Rede davon, dass Österreich im Triathlon der Herren vielleicht die erste Medaille holen könnte. Triathlon? Triathlon? ... Triathlon!

Sofort ist mir die Idee der drei Sportarten (Kompetenzen) Schwimmen, Radfahren und Laufen in meinen Kopf geschossen, die allesamt in einem bestimmten Verhältnis zu absolvieren sind. Übersichtlich, jeder hat ein Bild im Kopf, kurzum: perfekt geeignet!

Als ich aus meinen Tagträumen aufgewacht war, musste ich feststellen, dass Österreich nach dem Triathlon (wie erwartet) noch immer mit 0 Medaillen dastand. Dies sollte sich leider (aus der Sicht eines leidenschaftlichen und patriotischen Passivsportlers mit österreichischem Pass) bis zum bitteren Ende der Spiele nicht ändern.

Ich habe lange an dieser Arbeit geschrieben. Mein Perfektionismus hat mich dazu angetrieben, noch mehr in die Tiefe zu gehen. Mich dazu angetrieben, es noch besser, noch übersichtlicher und noch verständlicher zu machen. Nicht die kleinste Lücke zuzulassen.

Euphorie, Rückschlag, Ehrgeiz, Enttäuschung, Hoffnung, Frust, Aufbruch, Licht am Ende des Tunnels... Es war eine lange Achterbahnfahrt der Gefühle für mich.

Aber am Ende überwiegt die Freude über eine Arbeit, auf die ich stolz sein kann! Und die Erleichterung darüber, Sätze wie diese bald nicht mehr ertragen zu müssen: „Bist du schon Magister? Was, du bist immer noch nicht Magister? Wann wirst du denn endlich fertig? Gib endlich ab!“ Ich gebe zu, das hat an mir genagt. Zwischenzeitlich hat es auch viele Momente der Resignation gegeben.

Ich möchte mich an dieser Stelle ganz herzlich bei allen Menschen bedanken, die mir den Mut gegeben haben, mich den zahlreichen Herausforderungen zu stellen und hartnäckig zu bleiben! Nicht aufzugeben und nach vorne zu schauen, das Feuer in mir wieder zu entfachen.

Bei meinen Eltern, die mich darüber hinaus während meines Studiums finanziell unterstützt haben. Bei denen ich bis zum Ende des Studiums wohnen durfte. Die mir Kraft, Halt und Sicherheit gegeben haben und immer hinter mir gestanden sind.

Bei meiner Oma, die mich während meines Studiums ebenfalls stark unterstützt hat.

Bei meinem Freund Dr. Thomas Ledl, der mich auf den Boden zurückgeholt hat, als ich auf der Perfektionismuswolke zu entschweben drohte. Der mich mit seinen Motivationskünsten zu Höchstleistungen inspiriert hat.

Ich möchte mich auch bei meinen Probelesern (insbesondere bei Dr. Andreas Baierl) für ihre zahlreichen Anregungen und Hinweise bedanken.

Bei meinem Betreuer Univ.-Prof. Dr. Walter J. Gutjahr, der mir mit Rat und Tat zur Seite gestanden ist. Speziell Satz 33 wird mir lange in Erinnerung bleiben, der in Kapitel 5 eine fundamentale Bedeutung innehat. Zwischenzeitlich hat es so ausgesehen, als würde dieses wichtige Fundament wegbrechen, da sich die Beweisführung als schwieriger erwiesen hat als anfangs gedacht. Seine wertvollen Inputs haben mir den Weg zur Vollendung des Beweises drastisch geebnet.

Und bei allen Lehrveranstaltungsleiterinnen und -leitern, die mir das notwendige Wissen für diese Magisterarbeit vermittelt haben.

Das Schreiben dieser Arbeit war wie ein Durchschreiten eines Irrgartens. Ein Irrgarten voller Fallen. Und für die Überwindung einiger dieser Fallen habe ich mehr Zeit gebraucht, als mir lieb war. Viele Ideen haben auf den ersten Blick vielversprechend ausgesehen, sich aber später leider als dunkle Sackgassen erwiesen.

Aber ich habe den Ausgang gefunden!

Ich lade Sie herzlich dazu ein, werte Leserin und werter Leser, gemeinsam mit mir diesen Irrgarten zu beschreiten. Ich führe Sie sicher durch!

Daniel Obszelka, Bakk.

# Inhaltsverzeichnis

<b>1. Einführung und Übersicht</b>	<b>11</b>
1.1. Einführung in die Projektportfoliooptimierung . . . . .	11
1.2. Ziele dieser Arbeit . . . . .	13
1.3. Aufbau dieser Arbeit . . . . .	14
1.4. Notation . . . . .	15
1.4.1. Indizes . . . . .	15
1.4.2. Mengen . . . . .	15
1.4.3. Modellparameter . . . . .	16
1.4.4. Funktionen . . . . .	16
1.4.5. Variablen und variable Konstrukte . . . . .	17
1.4.6. Fixpunktstrukture . . . . .	17
1.4.7. Ableitungskürzel . . . . .	18
1.4.8. Algorithmennotation . . . . .	18
1.4.9. Sonstiges zur Notation . . . . .	18
1.5. Eine Seite Triathlon . . . . .	19
<b>2. Das Basismodell</b>	<b>21</b>
2.1. Über Projektklassen und Perioden . . . . .	21
2.2. Über Kompetenzen und effektiven Arbeitsaufwand . . . . .	21
2.3. Über Kompetenzscores . . . . .	23
2.4. Über Effizienz . . . . .	24
2.5. Über mittlere Effizienz . . . . .	25
2.6. Über Kompetenzentwicklung . . . . .	26
2.7. Über Erträge und die Zielfunktion . . . . .	28
2.8. Das Optimierungsproblem und Annahmen . . . . .	29
2.9. Analyse der Parameter und deren Zusammenhängen . . . . .	30
2.9.1. Parameter $a$ und $b$ der Lernfunktion $\varphi(\cdot)$ . . . . .	30
2.9.2. Das Zeitkonto $\sigma$ . . . . .	33
<b>3. Gleichverteilte Zufallslösungen</b>	<b>35</b>
3.1. Konstruktion einer Gleichverteilung auf $\mathcal{S}_t$ . . . . .	35
3.2. Konstruktion einer Gleichverteilung auf $\mathcal{S}$ . . . . .	37
<b>4. Kurzfristige lokale Optimierung</b>	<b>39</b>
4.1. Gradient . . . . .	39
4.2. Zulässige und verbessernde Richtungen . . . . .	42
4.3. Korrigierter Gradient . . . . .	44
4.4. Notwendige Bedingungen für lokale Optima . . . . .	48
4.5. Sattelpunkten entfliehen . . . . .	52
4.6. Linesearch . . . . .	53
4.7. Suche nach lokalen Optima . . . . .	54

<b>5. Langfristige lokale Optimierung</b>	<b>57</b>
5.1. Notation und Annahmen . . . . .	57
5.2. Gleichgewichtszustände des Kompetenzentwicklungssystems . . . . .	58
5.2.1. Stabilitätsbedingungen für Kompetenzen . . . . .	59
5.2.2. Zusammenhängende Grenzportfolios . . . . .	60
5.2.3. Skalierungsinvarianz und Monotonieeigenschaft . . . . .	62
5.2.4. Existenz von Gleichgewichtszuständen . . . . .	63
5.2.5. Eindeutigkeit von Gleichgewichtszuständen . . . . .	65
5.2.6. Konvergenz zum Gleichgewichtszustand . . . . .	75
5.2.7. Grenzgewinn und Lösungsliste . . . . .	77
5.2.8. Zusammenfassung und Beispiel . . . . .	78
5.3. Berechnung des Gleichgewichtszustands . . . . .	81
5.3.1. Das Gleichungssystem . . . . .	81
5.3.2. Die „Holzhackermethode“ . . . . .	83
5.4. Bestimmung lokaler Grenzoptima . . . . .	87
5.4.1. Motivation . . . . .	87
5.4.2. Pattern Search . . . . .	91
5.4.3. Ebenen . . . . .	94
5.5. Iterierte lokale Suche nach lokalen Grenzoptima . . . . .	100
<b>6. Mittelfristige lokale Optimierung</b>	<b>103</b>
6.1. Annahmen und Konventionen . . . . .	103
6.2. Kompromiss zwischen kurzfristiger und langfristiger Optimierung . . . . .	103
6.3. Bestimmung mittelfristiger lokaler Optima . . . . .	108
6.3.1. Grundidee . . . . .	108
6.3.2. Wahl der Startlösung . . . . .	110
6.3.3. Aufrollen der Lösung . . . . .	111
6.3.4. Rolling Horizon Algorithmus . . . . .	111
6.4. Einfluss der Parameter $k$ , $\delta$ und $z_1$ sowie der Startlösung $x^S$ . . . . .	112
6.5. Gleichgewichte und Konvergenz . . . . .	114
6.6. Abschließende Bemerkungen zur mittelfristigen Optimierung . . . . .	118
<b>7. Diskussion und Anregungen</b>	<b>119</b>
7.1. Umgang mit Ebenen . . . . .	119
7.2. Unsichere Projekterträge und Zinssätze . . . . .	122
7.2.1. Stochastisches Programm . . . . .	122
7.2.2. Der diskrete Fall . . . . .	122
7.2.3. Der kontinuierliche Fall . . . . .	124
7.2.4. Erwartetes Problem und der Value of Stochastic Solution . . . . .	125
7.3. Hinarbeiten auf einen bestimmten Moment . . . . .	126
7.4. Korrelation zwischen den Projektklassen stärker einbauen . . . . .	126
7.5. Gezieltes Training von Kompetenzen . . . . .	129

<b>A. Zusammenfassung auf Deutsch</b>	<b>131</b>
<b>B. Summary in English</b>	<b>133</b>
<b>C. Verwendete Programme und Anmerkungen zur Umsetzung</b>	<b>135</b>
<b>D. Verzeichnisse</b>	<b>137</b>
D.1. Abbildungsverzeichnis . . . . .	137
D.2. Tabellenverzeichnis . . . . .	138
D.3. Algorithmenverzeichnis . . . . .	139
D.4. Literatur . . . . .	140
<b>E. Lebenslauf</b>	<b>143</b>



# 1. Einführung und Übersicht

## 1.1. Einführung in die Projektportfoliooptimierung

Es geht um Projektportfoliooptimierung unter einem Kompetenzentwicklungsmodell. Uns stehen verschiedene Projekte zur Verfügung, die wir bearbeiten können. Ziel ist es, die verfügbare Arbeitszeit derart auf diese Projekte aufzuteilen, dass der Gesamtertrag maximal ist.

Dabei sind für die Erledigung der Projekte verschiedene Kompetenzen in unterschiedlichem Ausmaß erforderlich. Machen wir von einer Kompetenz oft Gebrauch, werden wir besser in ihr und können in weiterer Folge Projekte, welche diese Kompetenzen benötigen, schneller bearbeiten. Das gleiche gilt aber auch umgekehrt.

Diese Arbeit basiert auf [Gut11]. Dort wird ein Spezialfall des zu Grunde liegenden Modells ausführlich diskutiert und der allgemeine Fall aufgeschrieben. Wir wollen in dieser Arbeit den allgemeinen Fall genauer unter die Lupe nehmen.

Wir fassen in dieser Arbeit Projekte zu Projektklassen zusammen. Anstatt einzelne Angestellte zu betrachten, fassen wir die Mitarbeiter zu einem Angestelltenteam zusammen. Diese Vereinfachung ermöglicht bessere Resultate und eine einfachere Handhabung. Viele Ergebnisse lassen sich auf andere Fälle hin adaptieren.

Sofern die Kompetenzentwicklung ausgeblendet wird und die einzelnen Angestellten und Projekte einzeln betrachtet werden, sieht man sofort die Parallelen zum Schedulingproblem. In [Bas00] etwa wird eine R&D-Anwendung in der chemischen Industrie erläutert, wo die Mitarbeiter den Projekten zugewiesen werden, basierend auf deren Kompetenzen.

[TYD06] befasst sich mit einer Anwendung in der industriellen Fertigung, während sich [AC07] der Softwareentwicklung zuwendet. In [EM08] werden die Fähigkeiten der Mitarbeiter mit den Erfordernissen der einzelnen Aufgaben verglichen und so einander zugeordnet, dass die Summe der Distanzen minimiert wird.

Wir sehen, dass die Anwendungsmöglichkeiten dieses Modells oder ähnlicher Modelle vielseitig sind. Vom Projektmanagement (zB R&D) über die Generierung von optimalen Trainingsplänen (siehe diese Arbeit), sogar die Bestimmung optimaler Fruchtanbaufolgen in der Landwirtschaft ließen sich prinzipiell mit solchen Modellen angehen.

Viele Papers auf dem Gebiet der Projektportfoliooptimierung gehen von binären Entscheidungsvariablen aus, so etwa [GKR07], [GKR<sup>+</sup>08a], [GKR<sup>+</sup>08b] und [SKG09]. Dort kann die Entscheidung getroffen werden, ob ein Projekt bearbeitet werden soll oder nicht. Weiters wird hier der Konflikt zwischen unmittelbaren Erträgen und Investitionen in die Zukunft angesprochen, ein Thema, das wir auch in dieser Arbeit ansprechen werden.

In [YFIN06] wird eine zweistufige Optimierung gefahren: In erster Instanz wird entschieden, welche Projekte bearbeitet werden sollen und in zweiter Instanz werden die gewählten Projekte dann bestmöglich den Angestellten zugewiesen, basierend auf deren Fähigkeiten.

Daneben gibt es auch die Überlegung, kontinuierliche Entscheidungen zu treffen, also die Zeit beliebig auf die Projekte aufzuteilen, wie etwa in [LK02] oder [Gut11]. Auch wir werden uns in dieser Magisterarbeit mit dem kontinuierlichen Fall befassen.

Die Möglichkeiten der Modellierung des Lernens (ohne Vergessen) sind vielseitig. Einige von ihnen werden zB in [WS06], [NGM07], [PS07], [AGM07], [Fio07] oder [HK08a] angesprochen.

Dass Erlerntes auch wieder vergessen werden kann, wird etwa in [AE90] oder [DAE95] angesprochen. In [CE05] wird – ähnlich wie in dieser Arbeit – das Ausmaß des Kompetenzzuwachses an die in die Kompetenzen investierte Zeit gekoppelt und das Vergessen über eine Verlernkonstante geregelt.

In [HK08b] wird die Möglichkeit des Outsourcings bestimmter Aufgaben angesprochen. Dadurch können Aufgaben, die intern nur ineffizient bearbeitet werden können zu einem bestimmten Preis ausgelagert werden, was unter Umständen zu einer (noch) stärkeren Spezialisierung der eigenen Unternehmung führt. Wir werden in dieser Arbeit das Thema Outsourcing außen vor lassen.

Die Parallelen zur Wertpapierportfoliooptimierung sind durchaus vorhanden. Während aber in der Finanzapplikation auf Grund der unsicheren Preisentwicklung die Diversifikation als Schutzschild gegen zu großes Risiko herhält, beobachten wir in der Projektportfoliooptimierung oftmals den „Zwang“ zur Spezialisierung.

Der Trade-Off zwischen Spezialisierung, um die Kompetenzen bestmöglich zu entwickeln und Diversifikation auf Grund schwankender Erträge wird als „Innovator’s dilemma“ bezeichnet, das zB in [Chr03] diskutiert wird.

In [GR10] wird dem Thema Unsicherheit Platz eingeräumt, ebenso wie zB in [LS12], wo eine szenarienbasierte Portfolioselektion erfolgt. In dieser Arbeit betrachten wir die Erträge als deterministisch, ganz am Ende schneiden wir kurz den Fall an, dass die Erträge über die Zeit zufälligen Schwankungen ausgesetzt sind.

Viele Publikationen setzen sich mit der Bereitstellung von Algorithmen zur Lösung der Problemstellung auseinander. In [ADLL02] wird ein genetischer Algorithmus präsentiert, in [FAK11] eine Variante der Ant Colony Optimization. Wir werden in dieser Arbeit ein maßgeschneidertes Gradientenverfahren erläutern.

Die Langfristigkeit sowie Techniken zur Ermittlung von Lösungen mit langfristigem Optimalitätspotenzial sind noch relatives Neuland. Dies wird einen sehr großen Teil dieser Arbeit ausmachen.

## 1.2. Ziele dieser Arbeit

Basierend auf den zahlreichen Erkenntnissen, die wir über diese Problemstellung sammeln, widmen wir uns in dieser Arbeit der Herleitung und Bereitstellung von Algorithmen, welche uns bei der Wahl des „richtigen“ Projektportfolios unterstützen.

Dabei unterscheiden wir zwischen der kurzfristigen und der langfristigen Sichtweise, welche in unmittelbarem Gegensatz zueinander stehen. Langfristig gute Lösungen sind in der Regel kurzfristig schlecht und kurzfristig gute Lösungen sind meist langfristig unbefriedigend. Die Vereinbarung beider Sichtweisen ist ein weiteres Thema dieser Arbeit.

Ich habe mich bemüht, die Problemstellung übersichtlich und anschaulich zu präsentieren. Insbesondere bei der Untersuchung der Langfristigkeit sind einige spannende mathematische Resultate gelungen.

**Diese Arbeit richtet sich an AnwenderInnen:** Ihnen steht am Ende der Arbeit ein vollständig ausgestatteter Werkzeugkasten zur Lösung des hier dargestellten Projektportfoliooptimierungsproblems zur Verfügung! Ich hoffe, dass ich die für die Anwendung relevanten Algorithmen und Ideen übersichtlich und verständlich präsentieren kann!

**Diese Arbeit richtet sich an ForscherInnen:** Der Spielraum für Verbesserungen der Algorithmen ist groß und die Theorien und Ideen können ausgedehnt werden. Einige Aspekte habe ich im Anhang angeführt. Die Arbeit ist ein Mix aus Altbewährtem, das auf diese Problemstellung maßgeschneidert wird sowie neuen Ansätzen, die nach weiterer Forschung schreiben. Ich hoffe, dass ich Sie mit dieser Arbeit dazu inspirieren kann, sich ebenfalls mit diesem spannenden Thema auseinander zu setzen!

### 1.3. Aufbau dieser Arbeit

Diese Arbeit ist im Wesentlichen nach der eben erwähnten Sichtweise gegliedert. Die einzelnen Kapitel sind wie folgt aufgebaut:

**Kapitel 2 – Das Basismodell:** Wir leiten die Bestandteile der Problemstellung schrittweise her und fassen das Basismodell zusammen. Anschließend untersuchen wir einige ausgewählte Parameter auf deren Zusammenhänge.

**Kapitel 3 – Gleichverteilte Zufallslösungen:** Wir leiten einen Algorithmus her, der gleichverteilte Zufallslösungen auf der zulässigen Menge generiert.

**Kapitel 4 – Kurzfristige lokale Optimierung:** Wir stellen einen lokalen Optimierungsalgorithmus zur Lösung des Basismodells bereit, welcher auf Gradientenanstieg basiert. Die hierzu notwendigen Bausteine werden vorher der Reihe nach diskutiert.

**Kapitel 5 – Langfristige lokale Optimierung:** Was passiert, wenn wir unsere verfügbare Arbeitszeit immer gleich einsetzen? Wir untersuchen das Kompetenzentwicklungssystem auf Gleichgewichtszustände hin und diskutieren ihre Eindeutigkeit. Wir stellen einen Algorithmus zur Berechnung des Gleichgewichtszustands des Kompetenzentwicklungssystems bereit. Darauf aufbauend schreiben wir einen Algorithmus zur Bestimmung von Lösungen mit langfristigem Optimalitätspotenzial auf.

**Kapitel 6 – Mittelfristige lokale Optimierung:** Wir überlegen uns, wie sich die kurzfristige und langfristige Sichtweise vereinbaren lassen und formulieren einen entsprechenden Lösungsalgorithmus, der nach einem Kompromiss zwischen kurzfristiger und langfristiger Optimalität sucht.

**Kapitel 7 – Diskussion und Anregungen:** Weitere interessante Themen werden kurz angeschnitten. Unter anderem regen wir an, wie wir eine Menge von sehr ähnlichen Lösungen komprimieren können und wie wir mit unsicheren Projekterträgen und Zinssätzen umgehen können.

## 1.4. Notation

Um die Übersicht zu erhöhen, möchte ich an dieser Stelle die in dieser Arbeit verwendete Notation anführen. Dieser Abschnitt ist Ihr Stadtplan, der Sie sicher durch die Straßen und Gassen dieser Arbeit führen wird!

### 1.4.1. Indizes

$i, j$	Indizes für die Projektklassen
$n$	Anzahl der Projektklassen
$s, t$	Indizes für die Zeit
$k$	Horizontparameter beim Rolling Horizon Verfahren, sonst Allzweckwaffe
$T$	Anzahl der betrachteten Perioden, Zeithorizont
$T_1$	Legt fest, wie viele Perioden mit Rolling Horizon simuliert werden
$r, \rho$	Indizes für die Kompetenzen
$R$	Anzahl der Kompetenzen

### 1.4.2. Mengen

$\mathcal{E}$	Ebene, ein zusammenhängender Bereich von Grenzportfolios, dessen Grenzgewinne gleich sind, also $\bar{w}(\bar{x}) = \bar{w}(\bar{x}) \forall \bar{x}, \bar{x} \in \mathcal{E}$
$I_t(x)$	Indexmenge aller aktiven Nebenbedingungen für eine zulässige Lösung $x$ in Periode $t$ : $I_t(x) = \{i \in \mathcal{N} : x_{it} = 0\}$
$L$	Lösungsliste für Grenzportfolios, $L = L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$
$\mathcal{L}$	Liste von Lösungslisten
$\mathbb{N}/\mathbb{N}_0$	Menge der natürlichen Zahlen ohne/mit Null
$\mathcal{N}$	Indexmenge aller Projektklassen, $\mathcal{N} = \{1, \dots, n\}$
$\bar{\mathcal{N}}(\bar{x})$	Indexmenge aller Projektklassen, welche im Grenzportfolio $\bar{x}$ vorkommen, $\bar{\mathcal{N}}(\bar{x}) = \{i \in \mathcal{N} : \bar{x}_i > 0\}$
$\bar{\mathcal{N}}$	Sehr häufig verwendete Kurzschreibweise für $\bar{\mathcal{N}}(\bar{x})$
$\mathcal{R}$	Indexmenge aller Kompetenzen, $\mathcal{R} = \{1, \dots, R\}$
$\mathcal{R}_i$	Indexmenge aller Kompetenzen, welche für Projektklasse $i$ benötigt werden, $\mathcal{R}_i = \{r \in \mathcal{R} : d_{ir} > 0\}$
$\bar{\mathcal{R}}(\bar{x})$	Indexmenge aller Kompetenzen, welche für das Grenzportfolio $\bar{x}$ benötigt werden, $\bar{\mathcal{R}}(\bar{x}) = \cup_{i \in \bar{\mathcal{N}}(\bar{x})} \mathcal{R}_i$
$\bar{\mathcal{R}}$	Sehr häufig verwendete Kurzschreibweise für $\bar{\mathcal{R}}(\bar{x})$
$\bar{\mathcal{R}}_A$	Indexmenge aller Kompetenzen in $\bar{\mathcal{R}}$ , für die Stabilität A gilt, $\bar{\mathcal{R}}_A = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0\}$
$\bar{\mathcal{R}}_B$	Indexmenge aller Kompetenzen in $\bar{\mathcal{R}}$ , für die Stabilität B gilt, $\bar{\mathcal{R}}_B = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0\}$
$\bar{\mathcal{R}}_C$	Indexmenge aller Kompetenzen in $\bar{\mathcal{R}}$ , für die Stabilität C gilt, $\bar{\mathcal{R}}_C = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0\}$

$\mathcal{S}$	Zulässiger Bereich; Menge aller zulässigen Lösungen, $\mathcal{S} = \{u \in \mathbb{R}^{n \times T} : \sum_{i=1}^n u_{it} = \sigma_t \forall t \in \mathcal{T}, u \geq 0\} = \times_{t=1}^T \mathcal{S}_t$
$\bar{\mathcal{S}}$	Zulässiger Bereich für Grenzportfolios, $\bar{\mathcal{S}} = \{\bar{u} \in \mathbb{R}^n : \sum_{i=1}^n \bar{u}_i = \sigma, \bar{u} \geq 0\}$
$\mathcal{S}_t$	Menge aller zulässigen Teillösungen für Periode $t$ , $\mathcal{S}_t = \{u \in \mathbb{R}^n : \sum_{i=1}^n u_i = \sigma_t, u \geq 0\}$
$\mathcal{T}$	Indexmenge aller Perioden, $\mathcal{T} = \{1, \dots, T\}$
$\mathcal{T}_s^k$	Indexmenge aller Perioden, die zwischen $s$ und $s + k - 1$ liegen, $\mathcal{T}_s^k = \{s, s + 1, \dots, s + k - 1\}$

### 1.4.3. Modellparameter

$a$	Parameter der Lernfunktion $\varphi(\cdot)$ , $a > 0$
$\alpha_r$	Erlernkonstante von Kompetenz $r$ , $\alpha_r \geq 0$ ( $\alpha_r > 0$ für langfristige Studie)
$b$	Parameter der Lernfunktion $\varphi(\cdot)$ , $b > 0$
$\beta_r$	Verlernkonstante von Kompetenz $r$ , $\beta_r \geq 0$ ( $\beta_r > 0$ für langfristige Studie)
$d_{ir}$	Relativer effektiver Arbeitsaufwand; Anteil der in Projektklasse $i$ investierten Zeit, welcher im Falle perfekter Effizienzen in Arbeiten der $r$ . Kompetenz gesteckt werden muss, $d_{ir} \geq 0$ , $\sum_{r=1}^R d_{ir} = 1$
$\delta$	Aufzinsungsfaktor, $\delta \geq 1$
$\epsilon$	Genauigkeitsparameter, $\epsilon > 0$ klein
$\epsilon_0$	Wird in Kapitel 5 bei Stabilität C zur Annäherung von 0 benötigt, $\epsilon_0 > 0$ sehr klein, $\bar{\gamma}_r \geq \epsilon_0 \forall r \in \mathcal{R}$
$\sigma_t$	Verfügbares Zeitkonto in Periode $t$ , $\sigma_t > 0$
$\sigma$	Verfügbares Zeitkonto in jeder Periode im Falle konstanter Zeitkonten, $\sigma > 0$
$w_{it}$	Ertrag pro Nettoarbeitszeiteinheit für Projektklasse $i$ in Periode $t$ , $w_{it} > 0$ , Nettoarbeitszeit = $\eta_{it} x_{it}$
$w_i$	Ertrag pro Nettoarbeitszeiteinheit für Projektklasse $i$ im Falle konstanter Erträge, $w_i > 0$ , Nettoarbeitszeit = $\eta_{it} x_{it}$ bzw. $\bar{\eta}_i \bar{x}_i$ (Kapitel 5)
$z_{r1}$	Startkompetenzscore von Kompetenz $r$ , $z_{r1} \in \mathbb{R}$

### 1.4.4. Funktionen

$G(\cdot)$	Zielfunktion des Basismodells, $G(x) = \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it}$
$G_s^k(\cdot)$	Zielfunktion in Rolling Horizon, $G_s^k(x) = \sum_{t \in \mathcal{T}_s^k} \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it}$
$\bar{w}(\cdot)$	Grenzgewinn, $\bar{w}(\bar{x}) = \sum_{i=1}^n w_i \bar{\eta}_i \bar{x}_i$
$\varphi(\cdot)$	Lernfunktion, $\varphi(z) = [1 + a \cdot \exp(-b \cdot z)]^{-1}$ , $0 \leq \varphi(z) \leq 1$
$\bar{\varphi}(\cdot)$	Lernfunktion in Kapitel 5, $\bar{\varphi}(z) = \epsilon_0 + (1 - \epsilon_0) \cdot \varphi(z)$ , $\epsilon_0 \leq \bar{\varphi}(z) \leq 1$
$ \cdot $	Mächtigkeit (= Anzahl der Elemente in) der übergebenen Menge oder Absolutbetrag des übergebenen Skalars
$\ \cdot\ $	Euklidische Norm des übergebenen Vektors; berechnet die Summe der quadrierten Einträge des übergebenen Vektors
$\ \cdot\ _\infty$	Supremumnorm des übergebenen Vektors; gibt den größten absoluten Wert des übergebenen Vektors zurück

#### 1.4.5. Variablen und variable Konstrukte

$x_{it}$	Entscheidungsvariable; Zeit, die wir in Periode $t$ in Projektklasse $i$ investieren, $0 \leq x_{it} \leq \sigma_t$ und $\sum_{i=1}^n x_{it} = \sigma_t$
$x_t$	Teillösung für die Periode $t$ ; $x_t = (x_{1t}, x_{2t}, \dots, x_{nt})^T$
$x$	Matrixschreibweise für die Entscheidungsvariable, $x = [x_1, x_2, \dots, x_T]$
$\hat{x}$	Lokal optimale k-Schablone, siehe Seite 114
$z_{rt}$	Kompetenzscore von Kompetenz $r$ in Periode $t$ , wobei $z_{r1}$ gegeben ist, $z_{rt} = z_{rt-1} - \beta_r + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \eta_{it-1} x_{it-1}$ für $t \geq 2$
$d_{irt}$	Relativer tatsächlicher Arbeitsaufwand; Anteil der in Projektklasse $i$ investierten Zeit, welcher für die in Periode $t$ herrschenden Effizienzwerte in Arbeiten der $r$ . Kompetenz gesteckt werden muss, $d_{irt} = \frac{d_{ir}}{\gamma_{rt}} \eta_{it} \geq 0$ , $\sum_{r=1}^R d_{irt} = 1$
$\Delta z_{rt}$	Kompetenzscorezuwachs in Periode $t$ : $\Delta z_{rt} = z_{rt+1} - z_{rt}$
$\gamma_{rt}$	Effizienz von Kompetenz $r$ in Periode $t$ , $0 \leq \gamma_{rt} = \varphi(z_{rt}) \leq 1$
$\eta_{it}$	Effizienz, mit der Projektklasse $i$ in Periode $t$ bearbeitet werden kann, $0 \leq \eta_{it} = \left( \sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}} \right)^{-1} \leq 1$

Die Werte für  $z_{rt}$ ,  $d_{irt}$ ,  $\Delta z_{rt}$ ,  $\gamma_{rt}$  und  $\eta_{it}$  hängen allesamt von der zugrunde liegenden Lösung  $x$  ab. Wir verzichten auf die Möglichkeit, dies explizit anzuführen.

#### 1.4.6. Fixpunktstrukture

$\bar{x}$	Konstantes Grenzportfolio, $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)^T$ , $x_t = \bar{x} \forall t \geq 1$ , $\bar{x} \in \bar{\mathcal{S}}$
$\Delta \bar{z}_r$	Kompetenzscorezuwachs von Kompetenz $r$ im Fixpunkt, wobei $x_t = \bar{x} \forall t \geq 1$ , $\Delta \bar{z}_r = -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \bar{d}_{ir} \bar{x}_i$
$\bar{\gamma}_r$	Effizienz von Kompetenz $r$ im Fixpunkt, wobei $x_t = \bar{x} \forall t \geq 1$ , $\epsilon_0 \leq \bar{\gamma}_r \leq 1$
$\bar{\eta}_i$	Effizienz von Projektklasse $i$ im Fixpunkt, wobei $x_t = \bar{x} \forall t \geq 1$ , $\epsilon_0 \leq \bar{\eta}_i = \left( \sum_{r \in \mathcal{R}_i} \frac{d_{ir}}{\bar{\gamma}_r} \right)^{-1} \leq 1$
$\bar{d}_{ir}$	Relativer tatsächlicher Arbeitsaufwand im Fixpunkt, wobei $x_t = \bar{x} \forall t \geq 1$ . Anteil der in Projektklasse $i$ investierten Zeit, welcher für die im Fixpunkt herrschenden Effizienzen in Arbeiten der $r$ . Kompetenz gesteckt werden muss, $\bar{d}_{ir} = \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \geq 0$ , $\sum_{r \in \mathcal{R}_i} \bar{d}_{ir} = 1$

Wir verzichten auf die Möglichkeit, den Zusammenhang zum zugrunde liegenden Grenzportfolio  $\bar{x}$  explizit anzuführen! Für den Gleichgewichtszustand ist es im Prinzip ausreichend,  $\bar{\gamma} = (\bar{\gamma}_1, \dots, \bar{\gamma}_R)^T$  anzugeben, da sich alle anderen Konstrukte daraus ergeben. Die Ausdrücke  $\Delta \bar{z}_r$ ,  $\bar{\eta}_i$  und  $\bar{d}_{ir}$  sind allesamt Funktionen in  $\bar{\gamma}$ . Im Bedarfsfall notieren wir mit  $\Delta \bar{z}_r(\bar{\gamma})$ ,  $\bar{\eta}_i(\bar{\gamma})$  und  $\bar{d}_{ir}(\bar{\gamma})$  die Auswertung dieser Funktionen an der Stelle  $\bar{\gamma}$ .

### 1.4.7. Ableitungskürzel

$\nabla G(x)$  Gradient(enmatrix),  $\nabla G(x) = [\dot{G}_x^{js}]_{j \in \mathcal{N}, s \in \mathcal{T}}$

$\dot{G}_x$  Siehe  $\nabla G(x)$

$\ddot{G}_x$  Korrigierter Gradient; eine aus  $\dot{G}_x$  erstellte zulässige Verbesserungsrichtung

$$\dot{G}_x^{js} = \frac{\partial G(x)}{\partial x_{js}} \quad \eta_{it}^{js} = \frac{\partial \eta_{it}}{\partial x_{js}} \quad \dot{\gamma}_{rt}^{js} = \frac{\partial \gamma_{rt}^{-1}}{\partial x_{js}} \quad \dot{z}_{rt}^{js} = \frac{\partial z_{rt}}{\partial x_{js}}$$

### 1.4.8. Algorithmennotation

**Break** Bricht die aktuelle Schleife komplett ab

**Continue** Bricht aktuellen Schleifendurchlauf ab und setzt mit dem nächsten fort

**Null** Null-Objekt – ein Objekt ohne Inhalt

**Restart** Startet den Algorithmus neu

Weiters verwenden wir phasenweise aus Platzgründen den Strichpunkt. Dieser schließt einen Befehl ab, sodass wir mehrere Befehle in eine Zeile packen können.

Weiters wird für Lösungslisten  $L$  der Punktoperator benützt, der auf einzelne Komponenten von  $L$  zugreift. Beispiel:  $L.\bar{\gamma}$  greift aus der Liste  $L$  das Element  $\bar{\gamma}$  heraus.

### 1.4.9. Sonstiges zur Notation

- Für die Transposition von Vektoren und Matrizen verwenden wir  $T$ .  
ZB bezeichnet  $(\bar{\gamma}_1, \bar{\gamma}_2, \dots, \bar{\gamma}_R)^T$  den Spaltenvektor  $\bar{\gamma}$ .
- Wenn möglich bzw. nicht anders angegeben, bezeichnet ein Buchstabe ohne Index den Spaltenvektor mit seinen indizierten Werten.  
ZB  $\bar{\gamma} = (\bar{\gamma}_1, \bar{\gamma}_2, \dots, \bar{\gamma}_R)^T$ ,  $\beta = (\beta_1, \beta_2, \dots, \beta_R)^T$
- Matrizen, die mit nur einem Index versehen sind, deuten Spalten der Matrix an.  
ZB  $z_t = (z_{1t}, z_{2t}, \dots, z_{Rt})^T$ ,  $x_t = (x_{1t}, x_{2t}, \dots, x_{nt})^T$
- Hochgestellte Zahlen in Klammern (zB  $\gamma^{(k)}$ ) sind Iterationszähler.
- Der Zugriff auf Bereiche eines Vektors erfolgt durch Indizierung mit Indexmengen.  
ZB  $\bar{\gamma}_{\bar{\mathcal{R}}_A}$  entnimmt dem Vektor  $\bar{\gamma}$  jene Elemente, dessen Indizes in der Menge  $\bar{\mathcal{R}}_A$  enthalten sind.
- Der Zugriff auf mehrere Spalten einer Matrix erfolgt nach dem gleichen Schema.  
ZB  $x_{\mathcal{T}_s^k}$  entnimmt der Matrix  $x$  jene Spalten, dessen Indizes in der Menge  $\mathcal{T}_s^k$  enthalten sind.

## 1.5. Eine Seite Triathlon

Anhand des Triathlonbeispiels werden wir einen großen Teil der Theorie dieser Magisterarbeit veranschaulichen. Es ist aufbauend und wird regelmäßig erweitert. Wir listen hier die übliche Parameterkonstellation auf und präsentieren eine zentrale Grafik.

Auch wenn Sie im Moment (noch) nicht viel damit anfangen können (in diesem Fall haben Sie hoffentlich gerade erst mit dem Lesen begonnen), können Sie später diese Seite im Bedarfsfall aufsuchen, um einer Hin- und Herbällerei entfliehen zu können.

Die drei Disziplinen (Kompetenzen):

r = 1: **Schwimmen**  
 r = 2: **Radfahren**  
 r = 3: **Laufen**

Die drei Sportarten (Projektklassen)

i = 1: **Triathlon (alle Sportarten)**  
 i = 2: **Duathlon (Radfahren und Laufen)**  
 i = 3: **Laufen**

Die übliche Parameterkonstellation lautet:

$$d = \begin{bmatrix} 0.45 & 0.35 & 0.2 \\ 0 & 0.8 & 0.2 \\ 0 & 0 & 1 \end{bmatrix} \quad \beta = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix} \quad \alpha = \begin{pmatrix} 0.7 \\ 0.4 \\ 0.9 \end{pmatrix} \quad w = \begin{pmatrix} 300 \\ 200 \\ 100 \end{pmatrix} \quad \sigma = 1$$

Weiters umfasst eine Periode 3 Monate, die Parameter der Lernfunktion sind standardgemäß  $a = 1$  und  $b = 5$ . In Einzelfällen kann es Abweichungen von den eben angeführten Werten geben, diese sind dann aber gesondert angeschrieben.

Abbildung 1 zeigt für die übliche Parameterfestlegung die Grenzgewinne.

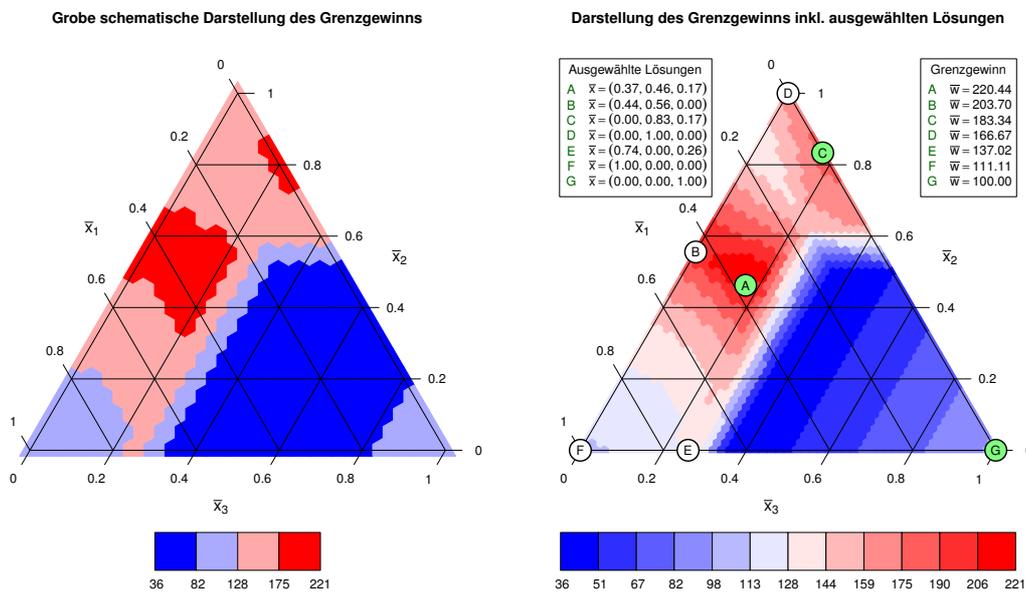


Abbildung 1: Grenzgewinne im Triathlonbeispiel in der üblichen Konstellation



## 2. Das Basismodell

Wir werden in diesem Kapitel das Modell herleiten. In den Abschnitten 2.1 bis 2.7 gehen wir dabei auf die einzelnen Bestandteile ein und in Abschnitt 2.8 fassen wir das Modell zusammen und stellen es übersichtlich dar. Schließlich untersuchen wir in Abschnitt 2.9 die Zusammenhänge einiger Parameter.

### 2.1. Über Projektklassen und Perioden

Eine Menge von  $n$  Projektklassen, indiziert mit  $i = 1, \dots, n$ , ist gegeben. Die Indexmenge aller Projektklassen bezeichnen wir mit  $\mathcal{N} = \{1, \dots, n\}$ . In einer Projektklasse werden mehrere Projekte des selben Typs zusammengefasst, zwischen welchen in unserem Modell nicht unterschieden wird.

Weiters wird die Zeit in  $T \in \mathbb{N}$  diskrete Perioden unterteilt, wobei der Planungshorizont  $T$  vorgegeben wird. In jeder Periode  $t \in \mathcal{T} = \{1, \dots, T\}$  steht uns ein Zeitkonto  $\sigma_t > 0$  zur Verfügung, das wir beliebig auf die Projektklassen aufteilen können.

Sei  $x_{it}$  die Zeit, die wir in Periode  $t$  in Projektklasse  $i$  investieren. Wir fassen die Entscheidungsvariablen  $x_{it}$  in eine Matrix  $x$  zusammen:

$$x := \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1T} \\ x_{21} & x_{22} & \dots & x_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nT} \end{bmatrix} = [x_1, x_2, \dots, x_T]$$

**Definition 1:** Eine Lösung  $x$  heißt **zulässig**, wenn folgende Bedingungen erfüllt sind:

$$\sum_{i=1}^n x_{it} = \sigma_t \quad \forall t \in \mathcal{T} \quad (1)$$

$$x_{it} \geq 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (2)$$

Wir schreiben oft auch einfach  $x \in \mathcal{S}$ , wobei  $\mathcal{S}$  der zulässige Bereich ist:

$$\mathcal{S} = \left\{ u \in \mathbb{R}^{n \times T} : \sum_{i=1}^n u_{it} = \sigma_t \quad \forall t \in \mathcal{T}, u \geq 0 \right\} \quad (3)$$

**Bemerkung 2:** Technisch gesehen müssten wir eigentlich  $\sum_{i=1}^n x_{it} \leq \sigma_t$  fordern, allerdings stellt sich später schnell heraus, dass die Bedingung (1) zielführender ist.

Falls  $\sigma_t = \sigma > 0 \quad \forall t \geq 1$  gilt, so sprechen wir von **konstanten Zeitkonten**.

### 2.2. Über Kompetenzen und effektiven Arbeitsaufwand

Wir betrachten eine Menge von  $R$  Kompetenzen, indiziert mit  $r = 1, \dots, R$ , die für die Bearbeitung der Projekte relevant sind. Wiederum bezeichnet  $\mathcal{R} = \{1, \dots, R\}$  die

Indexmenge aller Kompetenzen. Für die Projektklassen werden Kompetenzen in unterschiedlichem Ausmaß benötigt.

In der Regel haben die Kompetenzen unterschiedliche Skalenniveaus und/oder Maßeinheiten. Um die Bedarfsmengen der Kompetenzen vergleichbar zu gestalten, rechnen wir mit Zeitbedarfsmengen.

**Definition 3:** *Unter dem **relativen effektiven Arbeitsaufwand**  $d_{ir} \geq 0$  verstehen wir jenen relativen Zeitbedarf, der in Projektklasse  $i$  für Arbeiten der Kompetenz  $r$  anfällt, unter der Voraussetzung, dass jede Kompetenz mit der maximal möglichen Geschwindigkeit eingesetzt werden kann. Dieser ist normiert, das heißt es gilt:*

$$\sum_{r=1}^R d_{ir} = 1 \quad \forall i \in \mathcal{N} \quad (4)$$

Die Normierung ermöglicht es uns, Projekte mit unterschiedlicher Länge zusammenzufassen. Zwei Projekte gehören also genau dann zur selben Projektklasse, falls sie identisch sind bezüglich ihrer relativen effektiven Arbeitsaufwände.

**Beispiel 4:** *Wir betrachten zur Veranschaulichung das Triathlonbeispiel, das uns durch die gesamte Magisterarbeit begleiten wird.*

*Um das Projekt Triathlon-Kurzdistanz aus der Klasse Triathlon erfolgreich abzuschließen, müssen  $s_1 = 5.4$  km geschwommen,  $s_2 = 35$  km geradelt und  $s_3 = 8$  km gelaufen werden<sup>1</sup>. Es gibt also  $R = 3$  Kompetenzen, die wir in der Menge  $\mathcal{R} = \{1, 2, 3\}$  zusammenfassen.*

*Wir nehmen folgende Maximalgeschwindigkeiten  $v_r^{\max}$  an<sup>2</sup>:*

- Schwimmen: 6 km/h
- Radfahren: 50 km/h
- Laufen: 20 km/h

*Zwar sind alle Kompetenzen in der selben Maßeinheit (km) gegeben, allerdings sind die Skalenniveaus sehr unterschiedlich. In Anlehnung an  $\text{Zeit} = \text{Weg}/\text{Geschwindigkeit}$  können wir uns den absoluten effektiven Arbeitsaufwand  $q_r$  ausrechnen:*

$$q_r := \frac{s_r}{v_r^{\max}} \quad \forall r \in \mathcal{R}$$

*Der relative effektive Arbeitsaufwand  $d_r$  ist dann nur noch Formsache:*

$$d_r := \frac{q_r}{\sum_{\rho=1}^R q_\rho} \quad \forall r \in \mathcal{R}$$

*In Tabelle 1 auf der nächsten Seite ist die Situation zusammengefasst.*

*Der Triathlon ist also in schnellstens 2 Stunden ( $0.9 + 0.7 + 0.4$ ) zu absolvieren. Obwohl beim Schwimmen die wenigsten Kilometer zu absolvieren sind, wird – unter der Annahme,*

<sup>1</sup>Die Werte wurden gegenüber den offiziellen Distanzen geändert, damit schönere Zahlen herauskommen.

<sup>2</sup>Geschätzte und gerundete Geschwindigkeiten, um die Übersicht zu verbessern

$r$	Kompetenz	$s_r$	$v_r^{max}$	$q_r$	$d_r$
1	Schwimmen	5.4 km	6 km/h	0.9 h	0.45
2	Radfahren	35.0 km	50 km/h	0.7 h	0.35
3	Laufen	8.0 km	20 km/h	0.4 h	0.20

Tabelle 1: Das Projekt Triathlon-Kurzdistanz. Kompetenzen, Bedarfsmengen, maximale Geschwindigkeiten sowie absoluter und relativer effektiver Arbeitsaufwand

*dass wir in allen Disziplinen perfekt sind – fast die Hälfte der Zeit im Wasser verbracht, da das Schwimmtempo mit Abstand am langsamsten ist. Abbildung 2 vergleicht die Distanzen mit den relativen effektiven Arbeitsaufwänden.*

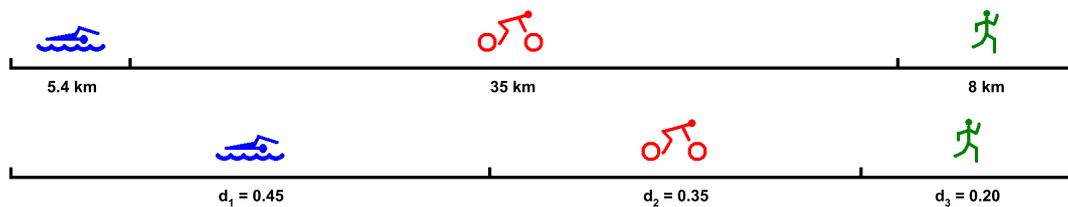


Abbildung 2: Darstellung des Projekts Triathlon-Kurzdistanz. Oben sind die Distanzen abgebildet und unten die relativen effektiven Arbeitsaufwände

*Wir betrachten in Tabelle 2 den Triathlon-Sprint mit halbierten Distanzen. Da die relativen effektiven Arbeitsaufwände mit jenen des Projekts Triathlon-Kurzdistanz übereinstimmen, gehören beide Projekte zur selben Projektklasse, nämlich Triathlon.*

$r$	Kompetenz	$s_r$	$v_r^{max}$	$q_r$	$d_r$
1	Schwimmen	2.7 km	6 km/h	0.45 h	0.45
2	Radfahren	17.5 km	50 km/h	0.35 h	0.35
3	Laufen	4.0 km	20 km/h	0.20 h	0.20

Tabelle 2: Das Projekt Triathlon-Sprint. Kompetenzen, Bedarfsmengen, maximale Geschwindigkeiten sowie absoluter und relativer effektiver Arbeitsaufwand

### 2.3. Über Kompetenzscores

Wir betrachten ein fixes Angestelltenteam. Zwischen den einzelnen Angestellten wird in unserem Modell nicht unterschieden, die Kompetenzwerte sind also firmenbezogen. Dies kann in erster Linie über eine Aggregation über alle Mitarbeiter erfolgen.

Eine wichtige und praxisrelevante Annahme in unserem Modell ist die Kompetenzentwicklung. Jede Kompetenz ist (er)lernbar bzw. ausbaufähig, sofern von ihr oft genug

Gebrauch gemacht wird. Umgekehrt entwickeln sich Kompetenzen zurück, die wenig bis gar nicht zur Anwendung kommen. In Abschnitt 2.6 erläutern wir, wie die Kompetenzentwicklung in unserem Modell genau funktioniert.

Das ist wohl ein günstiger Zeitpunkt, um die Kompetenzscores  $z_{rt} \in \mathbb{R}$  einzuführen. Sie messen, wie gut unser Unternehmen in Kompetenz  $r$  zum Zeitpunkt  $t$  ist. Je höher dieser Wert, desto schneller können wir Aufgaben, welche diese Kompetenz erfordern, meistern. Wir nehmen an, dass die Kompetenzscores entweder mittels psychologischer Methoden gemessen oder durch Erfahrung bestimmt werden können.

## 2.4. Über Effizienz

Ausgehend von den Kompetenzscores leiten wir den Effizienzwert  $\gamma_{rt}$  unter Zuhilfenahme einer monotonen Funktion  $\varphi : \mathbb{R} \mapsto (0, 1)$  ab.  $\gamma_{rt} = \varphi(z_{rt})$  modelliert die relative Geschwindigkeit, mit welcher wir Arbeiten in Kompetenz  $r$  zum Zeitpunkt  $t$  durchführen können. Gilt  $\gamma_{rt} = 1$ , so werden diese Arbeiten mit der maximal möglichen Geschwindigkeit verrichtet.

Die Lernfunktion ist also ein Modell für das Verhältnis von tatsächlicher Arbeitsgeschwindigkeit zu maximaler Arbeitsgeschwindigkeit. Wir verwenden die logistische Funktion, welche wie folgt definiert ist:

$$\varphi(z) = \frac{1}{1 + a \cdot \exp(-b \cdot z)} \quad (5)$$

Dabei sind  $a > 0$  und  $b > 0$  Parameter, welche die Lernkurve steuern. Abbildung 3 auf der nächsten Seite zeigt die Funktion für verschiedene  $a$  und  $b$ .

Die Lernfunktion lässt sich in zwei Abschnitte unterteilen: In einen konvexen Teil für niedrige Kompetenzscores und in einen konkaven Teil für hohe Kompetenzscores.

Wenn jemand kaum über Fertigkeiten in einer Kompetenz verfügt, so benötigt diese Person viel Zeit zum Erlernen der Grundkenntnisse. Sie muss sich erst in die Kompetenz einarbeiten und wird daher recht langsam Fortschritte erzielen.

Sobald aber die Grundlagen sitzen, beginnt jene Zeit, in der immer schneller Fortschritte erzielt werden können. Irgendwann flacht die Lernkurve jedoch wieder ab, da Perfektion nur mit äußerster Mühe erreicht werden kann.

Man kann zeigen, dass der Wendepunkt  $z^*$ , der beide Teile trennt, gegeben ist durch

$$z^* = \frac{-\ln(1/a)}{b} = \frac{\ln(a)}{b}$$

Dieser Wendepunkt in die Lernfunktion  $\varphi(\cdot)$  eingesetzt ergibt

$$\varphi(z^*) = \frac{1}{2}$$

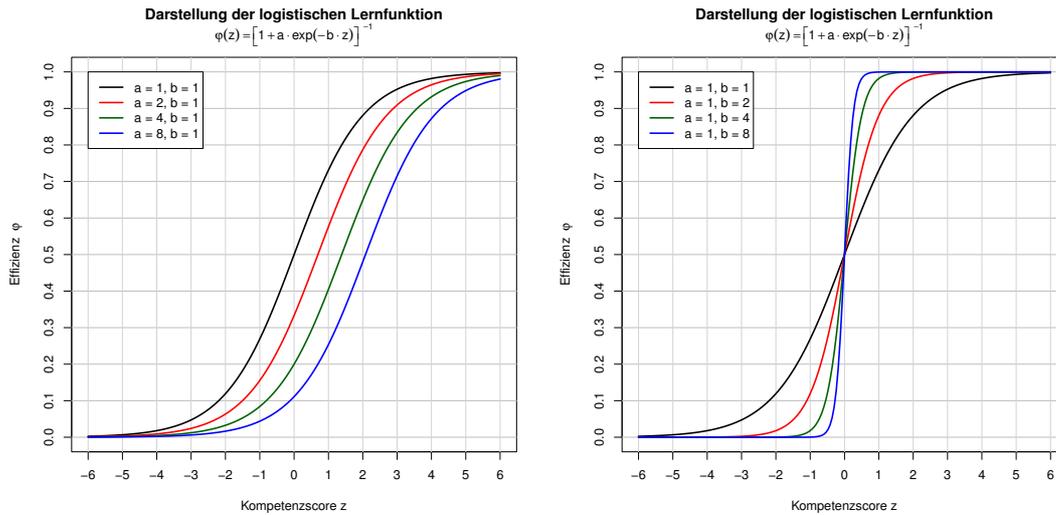


Abbildung 3: Veranschaulichung der Lernfunktion  $\varphi(\cdot)$  für verschiedene  $a$  und  $b$ .

**Links:** Der Parameter  $a$  verschiebt die Lernfunktion. **Rechts:** Der Parameter  $b$  steuert die Steilheit der Lernfunktion.

was wiederum bedeutet, dass der Lernfortschritt genau dann am größten ist, wenn die entsprechende Kompetenz mit der halben Effizienz eingesetzt werden kann.

## 2.5. Über mittlere Effizienz

Nun wollen wir die Effizienz von den Kompetenzen auf die Projektklassen ausdehnen.

**Definition 5:** Die mittlere Effizienz  $\eta_{it} \in [0, 1]$ , mit welcher Projektklasse  $i$  zum Zeitpunkt  $t$  bearbeitet werden kann, berechnet sich als harmonisches Mittel wie folgt:

$$\eta_{it} = \frac{\sum_{r=1}^R d_{ir}}{\sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}}} = \frac{1}{\sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}}} = \left( \sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}} \right)^{-1} \quad (6)$$

Es gibt viele Varianten, obige Formel zu interpretieren. Eine davon ist die folgende: Eine „perfekte“ Firma kann jede Kompetenz mit der maximalen Effizienz  $\gamma_{rt} = 1$  einsetzen. Je nach Länge eines beliebigen Projekts aus Projektklasse  $i$  wird diese Firma also  $c \cdot \sum_{r=1}^R d_{ir}$  Zeiteinheiten benötigen, um es fertig zu stellen. Wir hingegen benötigen für jede Kompetenz etwas länger, nämlich  $\frac{d_{ir}}{\gamma_{rt}}$  Zeiteinheiten, insgesamt also  $c \cdot \sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}}$  Zeiteinheiten. Der Quotient beider Größen ergibt die Effizienz für Projektklasse  $i$ .

**Beispiel 6:** Fortsetzung des Triathlon-Beispiels von Seite 22. Auf Seite 26 nochmals die Tabelle für Triathlon-Kurzstrecke, wobei  $v_r$  die tatsächliche Geschwindigkeit (in irgend-einer Periode) bezeichnet und die Effizienzwerte  $\gamma_r$  durch  $\frac{v_r}{v_r^{max}}$  gegeben sind.

$r$	Kompetenz	$s_r$	$v_r^{max}$	$v_r$	$\gamma_r$	$d_r$
1	Schwimmen	5.4 km	6 km/h	3 km/h	0.5	0.45
2	Radfahren	35.0 km	50 km/h	50 km/h	1.0	0.35
3	Laufen	8.0 km	20 km/h	16 km/h	0.8	0.20

Tabelle 3: Das Projekt Triathlon-Kurzdistanz revisited. Kompetenzen, Bedarfsmengen, maximale und tatsächliche Geschwindigkeiten sowie Effizienz und relativer effektiver Arbeitsaufwand

Wir werden anhand dieses Beispiels den Zusammenhang zwischen Geschwindigkeiten und Effizienz praktisch darstellen.

Zunächst berechnen wir die bestmögliche mittlere Geschwindigkeit:

$$\bar{v}^{max} = \frac{1}{\frac{5.4}{6} + \frac{35}{50} + \frac{8}{20}} = \frac{1}{0.9 + 0.7 + 0.4} = \frac{1}{2}$$

Es kann also pro Stunde bestenfalls ein halber Triathlon-Kurzdistanz absolviert werden. Analog die tatsächliche mittlere Geschwindigkeit:

$$\bar{v} = \frac{1}{\frac{5.4}{3} + \frac{35}{50} + \frac{8}{16}} = \frac{1}{1.8 + 0.7 + 0.5} = \frac{1}{3}$$

Die Effizienz ist das Verhältnis zwischen tatsächlicher und bestmöglicher Geschwindigkeit:

$$\eta = \frac{\bar{v}}{\bar{v}^{max}} = \frac{1/3}{1/2} = \frac{2}{3}$$

Die ineffiziente Schwimmleistung sorgt dafür, dass wir insgesamt nur eine Effizienz von  $2/3$  erreichen.

Nun die selbe Rechnung mit Definition 5:

$$\eta = \frac{1}{\frac{0.45}{0.5} + \frac{0.35}{1} + \frac{0.2}{0.8}} = \frac{1}{0.9 + 0.35 + 0.25} = \frac{1}{1.5} = \frac{2}{3}$$

Beispiel 6 zeigt, warum wir nicht mit Geschwindigkeiten sondern mit Effizienzen rechnen. So hätte der Triathlon-Sprint mit halbierten Distanzen eine doppelt so hohe Geschwindigkeit – pro Stunde kann ein ganzer Triathlon-Sprint im besten Fall absolviert werden.

Das gilt allgemein: Verschiedene Projekte einer Projektklasse sind in der Regel unterschiedlich aufwändig, die Effizienz ist jedoch immer ident.

## 2.6. Über Kompetenzentwicklung

Eine wichtige Annahme ist, dass die Kompetenzen gleichmäßig bei der Bearbeitung der Projekte einer Projektklasse eingesetzt werden müssen. Die folgende Definition wird uns bei der Klärung der Frage, was genau das heißen soll, behilflich sein.

**Definition 7:** Unter dem *relativen tatsächlichen Arbeitsaufwand*  $d_{irt} \in [0, 1]$  verstehen wir jenen relativen Zeitbedarf, der in Periode  $t$  in Projektklasse  $i$  für Arbeiten der Kompetenz  $r$  anfällt, unter der Voraussetzung, dass jede Kompetenz mit der tatsächlichen Geschwindigkeit eingesetzt wird. Dieser berechnet sich wie folgt:

$$d_{irt} = \frac{\frac{d_{ir}}{\gamma_{rt}}}{\sum_{\rho=1}^R \frac{d_{i\rho}}{\gamma_{\rho t}}} = \frac{d_{ir}}{\gamma_{rt}} \eta_{it} \quad (7)$$

Klarerweise gilt:

$$\sum_{r=1}^R d_{irt} = 1 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (8)$$

Im Gegensatz zum relativen effektiven Arbeitsaufwand  $d_{ir}$  variiert sein tatsächliches Pendant  $d_{irt}$  im Laufe der Zeit, je nachdem wie sich die Kompetenzen entwickeln.

Wie nehmen an, dass die in Projektklasse  $i$  investierte Zeit  $x_{it}$  proportional zu  $\frac{d_{ir}}{\gamma_{rt}}$  auf die Kompetenzen aufgeteilt werden. Diese Annahme garantiert uns, dass die einzelnen Projekte einer Projektklasse der Reihe nach zur Gänze abgearbeitet werden, anstatt eine Vielzahl halbfertiger Projekte anzuhäufen. Abbildung 4 zeigt die Bedeutung dieser Annahme grafisch auf.

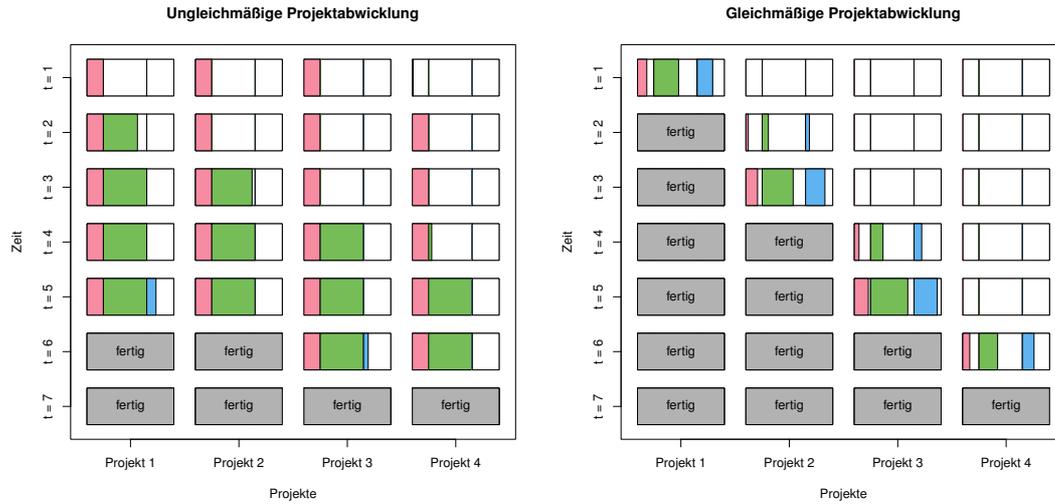


Abbildung 4: Unterschiedliche Projektentwicklungspolitiken. **Links:** Es häufen sich viele halbfertige Projekte an. **Rechts:** Ein Projekt nach dem anderen wird abgearbeitet.

Jetzt schauen wir uns an, nach welcher Regel sich die Kompetenzscores in unserem Modell entwickeln. Ausgehend von den Startkompetenzscores  $z_{r1} \in \mathbb{R}$ , welche gegeben sind, führen wir folgende rekursive Formel ein:

$$z_{rt} = z_{rt-1} - \beta_r + \alpha_r \underbrace{\sum_{i=1}^n d_{irt-1} x_{it-1}}_{\text{Kompetenzzuwachs}} \quad \forall r \in \mathcal{R}, t \geq 2 \quad (9)$$

Dabei sind  $\alpha_r \geq 0$  und  $\beta_r \geq 0$  Parameter.

Ausgehend vom Kompetenzscore der vergangenen Periode subtrahieren wir eine Konstante  $\beta_r$ . Dies simuliert das Vergessen: Wenn wir eine Kompetenz nie einsetzen, wird sie irgendwann einmal vergessen sein.

Beim Kompetenzzuwachs summieren wir die gesamte Zeit, welche wir in Periode  $t - 1$  mit Kompetenz  $r$  verbracht haben. Hier fließt die Annahme der gleichmäßigen Projektabwicklung ein. Dies wird noch mit dem Erlernfaktor  $\alpha_r$  multipliziert.

## 2.7. Über Erträge und die Zielfunktion

Die Bruttoarbeitszeit  $x_{it}$  ist jene Zeit, die wir in Periode  $t$  in Projektklasse  $i$  investieren. Von dieser Zeit wird aber nur der Anteil  $\eta_{it}$  genützt, der Rest verpufft infolge von Ineffizienzen. Somit ergibt sich aus der Bruttoarbeitszeit  $x_{it}$  die Nettoarbeitszeit  $\eta_{it} \cdot x_{it}$ .

**Definition 8:** Die Größe

$$\eta_{it} \cdot x_{it} = \left( \sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}} \right)^{-1} x_{it} \quad (10)$$

bezeichnen wir als **Nettoarbeitszeit** von Projektklasse  $i$  in Periode  $t$ .

Wenn wir uns schon die Mühe machen, Projekte zu bearbeiten, dann sollte es sich für uns auch lohnen! Jede Einheit der Nettoarbeitszeit  $\eta_{it} \cdot x_{it}$  wird mit einem Betrag  $w_{it} > 0$  vergütet. Zusätzlich werden zukünftige Erträge unter Zuhilfenahme des Aufzinsungsfaktors  $\delta \geq 1$  abgezinst, den wir zumeist kurz Zinssatz nennen.

Das führt uns nun zur Zielfunktion, die zu maximieren unser Bestreben darstellt:

$$G(x) = \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} \quad (11)$$

## 2.8. Das Optimierungsproblem und Annahmen

Wir fassen nun das Basismodell zusammen. Die Mengen im Überblick:

- $\mathcal{N} = \{1, 2, \dots, n\}$  – die Indexmenge der Projektklassen
- $\mathcal{R} = \{1, 2, \dots, R\}$  – die Indexmenge der Kompetenzen
- $\mathcal{T} = \{1, 2, \dots, T\}$  – die Indexmenge der Perioden

Das Optimierungsproblem:

$$G(x) = \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} \rightarrow \max! \quad (12)$$

wobei

$$z_{rt} = z_{rt-1} - \beta_r + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \eta_{it-1} x_{it-1} \quad \forall r \in \mathcal{R}, t \geq 2 \quad (13)$$

$$\gamma_{rt} = \varphi(z_{rt}) = \frac{1}{1 + a \cdot \exp(-b \cdot z_{rt})} \quad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad (14)$$

$$\eta_{it} = \left( \sum_{\rho=1}^R \frac{d_{i\rho}}{\gamma_{\rho t}} \right)^{-1} \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (15)$$

$$\sum_{i=1}^n x_{it} = \sigma_t \quad \forall t \in \mathcal{T} \quad (16)$$

$$x_{it} \geq 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (17)$$

### Annahmen und Voraussetzungen:

- Die Erträge  $w_{it} > 0$  und Zeitkonten  $\sigma_t > 0$  sind deterministisch.
- In jeder Projektklasse stehen stets genügend Projekte zur Verfügung. Die Kompetenzen werden bei der Projektabwicklung proportional zu  $d_{irt} = \frac{d_{ir}}{\gamma_{rt}} \eta_{it}$  eingesetzt.
- Die Startkompetenzscores  $z_{r1} \in \mathbb{R}$  sind gegeben.
- Die relativen effektiven Arbeitsaufwände  $d_{ir}$  können gemessen werden und erfüllen  $d_{ir} \geq 0$  sowie  $\sum_{r=1}^R d_{ir} = 1 \quad \forall i \in \mathcal{N}$
- Die Parameter der Lernfunktion erfüllen  $a > 0$  und  $b > 0$ .
- Die Lernkonstanten erfüllen  $\alpha_r \geq 0$  und  $\beta_r \geq 0$ .
- Der Aufzinsungsfaktor (Zinssatz) ist fix und es gilt  $\delta \geq 1$ .

Wir werden zu Beginn der folgenden Kapitel bei Bedarf zusätzliche Annahmen fällen. Die Indizes  $i$  und  $j$  verwenden wir für Projektklassen,  $r$  und  $\rho$  für Kompetenzen und  $s$  und  $t$  deuten Perioden an.

## 2.9. Analyse der Parameter und deren Zusammenhängen

Das Modell, wie wir es soeben aufgeschrieben haben, ist überparametrisiert. Wir werden feststellen, dass wir auf einige Parameter verzichten könnten, ohne die Problemstellung zu verändern. Zwecks besserer Übersicht werden wir aber alle Parameter behalten.

### 2.9.1. Parameter $a$ und $b$ der Lernfunktion $\varphi(\cdot)$

Wir werden feststellen, dass wir die Lernfunktion  $\varphi(\cdot)$  normieren können, wenn wir adäquat an den Schrauben anderer Parameter drehen. Die folgenden zwei allgemein formulierten Sätze geben uns eine Idee davon, wie.

**Satz 9:** Sei  $a > 0$  der originale Parameter der Lernfunktion  $\varphi(\cdot)$ , der auf  $\tilde{a} > 0$  geändert werden soll. Dann kann mit der Transformation

$$z_{r1} \leftarrow z_{r1} - \frac{\ln(a/\tilde{a})}{b} \quad \forall r \in \mathcal{R} \quad (18)$$

das ursprüngliche Problem reproduziert werden.

*Beweis.* Für ein beliebiges  $r \in \mathcal{R}$  sei  $\tilde{z}_{rt}$  der gesuchte neue Kompetenzscore in Periode  $t$  für die geänderte Lernfunktion

$$\tilde{\varphi}(z) = (1 + \tilde{a} \cdot \exp(-b \cdot z))^{-1}$$

Um das Problem nicht zu verändern, muss folgende Äquivalenz herrschen:

$$\tilde{\varphi}(\tilde{z}_{rt}) = \frac{1}{1 + \tilde{a} \cdot e^{-b \cdot \tilde{z}_{rt}}} \stackrel{(!)}{=} \frac{1}{1 + a \cdot e^{-b \cdot z_{rt}}} = \varphi(z_{rt})$$

Diese Gleichheit ist genau dann erfüllt, wenn gilt

$$\begin{aligned} \tilde{a} \cdot e^{-b \cdot \tilde{z}_{rt}} = a \cdot e^{-b \cdot z_{rt}} &\Leftrightarrow \ln \tilde{a} - b \cdot \tilde{z}_{rt} = \ln a - b \cdot z_{rt} \\ &\Leftrightarrow -b \cdot \tilde{z}_{rt} = \ln a - \ln \tilde{a} - b \cdot z_{rt} \\ &\Leftrightarrow \tilde{z}_{rt} = z_{rt} - \frac{1}{b} (\ln a - \ln \tilde{a}) \Leftrightarrow \tilde{z}_{rt} = z_{rt} - \frac{\ln(a/\tilde{a})}{b} \end{aligned}$$

Wegen (9) genügt es, diese additive Änderung in der ersten Periode vorzunehmen, also lediglich die Startkompetenzscores zu adaptieren. Da diese Überlegungen für jedes  $r \in \mathcal{R}$  gelten, haben wir den Satz bewiesen.  $\square$

**Satz 10:** Sei  $b > 0$  der originale Parameter der Lernfunktion  $\varphi(\cdot)$ , der auf  $\tilde{b} > 0$  geändert werden soll. Dann kann das ursprüngliche Problem mit den folgenden Transformationen zurückgewonnen werden.

$$z_{r1} \leftarrow z_{r1} \cdot b/\tilde{b} \quad \forall r \in \mathcal{R} \quad (19)$$

$$\alpha_r \leftarrow \alpha_r \cdot b/\tilde{b} \quad \forall r \in \mathcal{R} \quad (20)$$

$$\beta_r \leftarrow \beta_r \cdot b/\tilde{b} \quad \forall r \in \mathcal{R} \quad (21)$$

*Beweis.* Für ein beliebiges  $r \in \mathcal{R}$  sei analog wie im Beweis des vorangehenden Satzes  $\tilde{z}_{rt}$  der gesuchte neue Kompetenzscore in Periode  $t$  für die geänderte Lernfunktion

$$\tilde{\varphi}(z) = (1 + a \cdot \exp(-\tilde{b} \cdot z))^{-1}$$

Um das Problem nicht zu verändern, muss folgende Äquivalenz herrschen:

$$\tilde{\varphi}(\tilde{z}_{rt}) = \frac{1}{1 + a \cdot e^{-\tilde{b} \cdot \tilde{z}_{rt}}} \stackrel{(!)}{=} \frac{1}{1 + a \cdot e^{-b \cdot z_{rt}}} = \varphi(z_{rt})$$

Offensichtlich ist diese Gleichheit genau dann erfüllt, wenn gilt

$$\tilde{b} \cdot \tilde{z}_{rt} = b \cdot z_{rt} \quad \Leftrightarrow \quad \tilde{z}_{rt} = z_{rt} \cdot \frac{b}{\tilde{b}}$$

Wir verknüpfen diese Information für  $t \geq 2$  mit Gleichung (9), welche das Kompetenzentwicklungsmodell beschreibt:

$$\begin{aligned} \tilde{z}_{rt} &= \frac{b}{\tilde{b}} \cdot z_{rt} \\ &= \frac{b}{\tilde{b}} \cdot z_{rt-1} - \frac{b}{\tilde{b}} \cdot \beta_r + \frac{b}{\tilde{b}} \cdot \alpha_r \sum_{i=1}^n d_{irt-1} x_{it-1} \\ &= \tilde{z}_{rt-1} - \tilde{\beta}_r + \tilde{\alpha}_r \sum_{i=1}^n d_{irt-1} x_{it-1} \end{aligned}$$

Dabei sind

$$\begin{aligned} \tilde{\alpha}_r &= \alpha_r \cdot b/\tilde{b} \\ \tilde{\beta}_r &= \beta_r \cdot b/\tilde{b} \end{aligned}$$

die adaptierten Parameter der Kompetenzentwicklung. Für den Startkompetenzscore  $z_{r1}$  gilt schlicht:

$$\tilde{z}_{r1} = \frac{b}{\tilde{b}} \cdot z_{r1}$$

Da diese Überlegungen für jedes  $r \in \mathcal{R}$  gelten, haben wir den Satz bewiesen.  $\square$

Insbesondere die Normierung  $\tilde{a} = 1$  und  $\tilde{b} = 1$  ist von Interesse. In diesem Fall führen wir die Transformationen

$$z_{r1} \leftarrow b \cdot z_{r1} - \ln a \quad \forall r \in \mathcal{R} \quad (22)$$

$$\alpha_r \leftarrow \alpha_r \cdot b \quad \forall r \in \mathcal{R} \quad (23)$$

$$\beta_r \leftarrow \beta_r \cdot b \quad \forall r \in \mathcal{R} \quad (24)$$

durch und erhalten damit das ursprüngliche Problem zurück.

**Beispiel 11:** Wir veranschaulichen in diesem Beispiel mit einer Projektklasse und zwei Kompetenzen obige Resultate. Beide Kompetenzen haben den selben relativen effektiven Arbeitsaufwand, dh  $d_1 = d_2 = 1/2$ . In der linken Spalte befinden sich die Berechnungen für die originale und in der rechten jene für die normierte Parametrisierung. Ausgehend von den Startkompetenzscores rechnen wir uns für beide Varianten die Scores der 2. Periode aus und überprüfen, ob die entsprechenden Effizienzwerte tatsächlich übereinstimmen.

**Ursprüngliches Problem**

**Normiertes Problem**

Die ursprünglichen Werte für  $a$ ,  $b$ ,  $\alpha_r$ ,  $\beta_r$  und die Startkompetenzscores:

Normierung von  $a$  und  $b$  bzw. Transformationen gemäß (22) - (24):

$$\begin{aligned} \sigma &= 1 & z_1 &= (-2, 0)^T \\ a &= 2 & \alpha &= (4, 9)^T \\ b &= \ln 2 & \beta &= (2, 1/4)^T \end{aligned}$$

$$\begin{aligned} \tilde{\sigma} &= 1 & \tilde{z}_1 &= \ln 2 \cdot (-3, -1)^T \\ \tilde{a} &= 1 & \tilde{\alpha} &= \ln 2 \cdot (4, 9)^T \\ \tilde{b} &= 1 & \tilde{\beta} &= \ln 2 \cdot (2, 1/4)^T \end{aligned}$$

Effizienzwerte in Periode 1:

Effizienzwerte in Periode 1:

$$\begin{aligned} \gamma_{11} &= \varphi(z_{11}) = \frac{1}{1 + 2 \cdot e^{-\ln 2 \cdot (-2)}} = \frac{1}{9} \\ \gamma_{21} &= \varphi(z_{21}) = \frac{1}{1 + 2 \cdot e^{-\ln 2 \cdot 0}} = \frac{1}{3} \end{aligned}$$

$$\begin{aligned} \tilde{\gamma}_{11} &= \varphi(\tilde{z}_{11}) = \frac{1}{1 + e^{-(-3 \cdot \ln 2)}} = \frac{1}{9} \\ \tilde{\gamma}_{21} &= \varphi(\tilde{z}_{21}) = \frac{1}{1 + e^{-(-\ln 2)}} = \frac{1}{3} \end{aligned}$$

Wir schauen sogleich in die 2. Periode:

Und der Blick in die 2. Periode:

$$\begin{aligned} \eta_1 &= \left( \frac{1/2}{1/9} + \frac{1/2}{1/3} \right)^{-1} = \frac{1}{6} \\ z_{12} &= -2 - 2 + 4 \cdot \frac{1/2}{1/9} \cdot \frac{1}{6} = -1 \\ z_{22} &= 0 - \frac{1}{4} + 9 \cdot \frac{1/2}{1/3} \cdot \frac{1}{6} = 2 \\ \gamma_{12} &= \varphi(z_{12}) = \frac{1}{1 + 2 \cdot e^{-\ln 2 \cdot (-1)}} = \frac{1}{5} \\ \gamma_{22} &= \varphi(z_{22}) = \frac{1}{1 + 2 \cdot e^{-\ln 2 \cdot 2}} = \frac{2}{3} \end{aligned}$$

$$\begin{aligned} \tilde{\eta}_1 &= \left( \frac{1/2}{1/9} + \frac{1/2}{1/3} \right)^{-1} = \frac{1}{6} \\ \tilde{z}_{12} &= \ln 2 \left( -3 - 2 + 4 \cdot \frac{1/2}{1/9} \cdot \frac{1}{6} \right) = -2 \ln 2 \\ \tilde{z}_{22} &= \ln 2 \left( -1 - \frac{1}{4} + 9 \cdot \frac{1/2}{1/3} \cdot \frac{1}{6} \right) = \ln 2 \\ \tilde{\gamma}_{12} &= \tilde{\varphi}(z_{12}) = \frac{1}{1 + e^{-(-2 \cdot \ln 2)}} = \frac{1}{5} \\ \tilde{\gamma}_{22} &= \tilde{\varphi}(z_{22}) = \frac{1}{1 + e^{-\ln 2}} = \frac{2}{3} \end{aligned}$$

Vergleiche die Effizienzwerte mit jenen des ... und siehe da, sie stimmen allesamt überein!

In Abbildung 5 auf der nächsten Seite sind beide Lernfunktionen abgebildet. Zusätzlich stellen wir die Transformationen für die 2. Kompetenz grafisch dar.

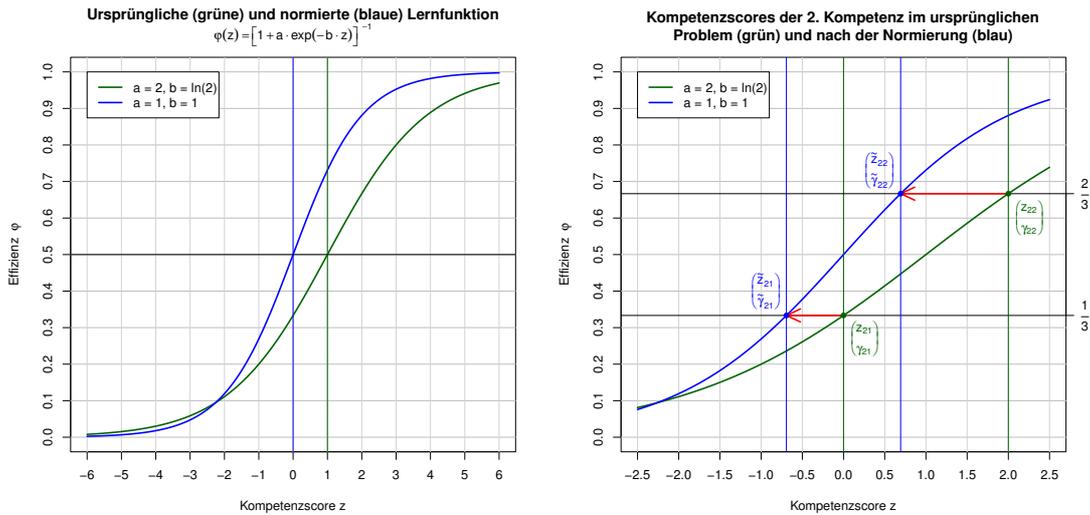


Abbildung 5: Transformation der Lernfunktionen in Beispiel 11. **Links:** Die Wendepunkte der ursprünglichen bzw. normierten Lernfunktion treten bei  $z = 1$  bzw.  $z = 0$  auf. **Rechts:** Veranschaulichung obiger Rechnung für die 2. Kompetenz. Die Kompetenzscores sind derart transformiert, dass die entsprechenden Effizienzwerte gleich sind, bzw. das Problem nicht verändert wird.

## 2.9.2. Das Zeitkonto $\sigma$

Im Falle konstanter Zeitkonten (das heißt  $\sigma_t = \sigma \forall t \geq 1$ ) könnten wir mit Hilfe des folgenden Satzes für eine weitere Parameterreduktion sorgen.

**Satz 12:** Sei  $\sigma > 0$  der originale Parameter für das konstante Zeitkonto, der auf  $\tilde{\sigma} > 0$  geändert werden soll. Dann kann das ursprüngliche Problem mit den Transformationen

$$\alpha_r \leftarrow \alpha_r \cdot \sigma / \tilde{\sigma} \quad \forall r \in \mathcal{R} \quad (25)$$

$$x_{it} \leftarrow x_{it} \cdot \tilde{\sigma} / \sigma \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (26)$$

zurückgewonnen werden. Die Zulässigkeitsbedingung (1) für das modifizierte Portfolio  $x$  lautet fortan

$$\sum_{i=1}^n x_{it} = \tilde{\sigma} \quad (27)$$

*Beweis.* Wir gehen für beliebiges  $t \in \mathcal{T}$ ,  $r \in \mathcal{R}$  von der Kompetenzentwicklungsgleichung (9) aus.

$$\begin{aligned}
z_{rt+1} &= z_{rt} - \beta_r + \alpha_r \sum_{i=1}^n d_{irt} x_{it} \\
&= z_{rt} - \beta_r + \alpha_r \sum_{i=1}^n d_{irt} x_{it} \cdot \frac{\tilde{\sigma}}{\sigma} \cdot \frac{\sigma}{\tilde{\sigma}} \\
&= z_{rt} - \beta_r + \alpha_r \cdot \frac{\sigma}{\tilde{\sigma}} \cdot \sum_{i=1}^n d_{irt} x_{it} \cdot \frac{\tilde{\sigma}}{\sigma} \\
&= z_{rt} - \beta_r + \tilde{\alpha}_r \sum_{i=1}^n d_{irt} \tilde{x}_{it}
\end{aligned}$$

Dabei verwenden wir die folgenden Adaptionen, um das Problem beizubehalten:

$$\begin{aligned}
\tilde{\alpha}_r &= \alpha_r \cdot \sigma / \tilde{\sigma} \\
\tilde{x}_{it} &= x_{it} \cdot \tilde{\sigma} / \sigma \quad \forall i \in \mathcal{N}
\end{aligned}$$

Die neue Zulässigkeitsbedingung lautet

$$\sum_{i=1}^n \tilde{x}_{it} = \tilde{\sigma} \quad \Leftrightarrow \quad \sum_{i=1}^n x_{it} = \sigma$$

Da diese Überlegungen für jedes  $t \in \mathcal{T}$  und  $r \in \mathcal{R}$  gelten, haben wir den Satz bewiesen.  $\square$

Auch hier ist die Normierung  $\tilde{\sigma} = 1$  von besonderem Interesse. In diesem Fall sind folgende Transformationen notwendig:

$$\alpha_r \leftarrow \alpha_r \cdot \sigma \quad \forall r \in \mathcal{R} \tag{28}$$

$$x_{it} \leftarrow x_{it} / \sigma \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{29}$$

Die neue Zulässigkeitsbedingung lautet dann

$$\sum_{i=1}^n x_{it} = 1 \quad \forall t \in \mathcal{T} \tag{30}$$

Folglich kann  $x_{it}$  als jener Anteil der Arbeitszeit in Periode  $t$  interpretiert werden, welcher in Projektklasse  $i$  gesteckt wird.

### 3. Gleichverteilte Zufallslösungen

Bevor wir mit der Optimierung anfangen, wollen wir uns der Konstruktion von zufälligen gleichverteilten Lösungen widmen. Ohne nähere Vorkenntnisse zu besitzen, ist es immer legitim, zunächst zufällige Startlösungen einer lokalen Maximierung zuzuführen. Außerdem greifen wir in weiterer Folge bei iterierten lokalen Suchverfahren auf zufällige Startlösungen zurück.

Ziel ist es, eine Gleichverteilung auf der zulässigen Menge  $\mathcal{S}$  zu konstruieren. Bevor wir starten, zerlegen wir den zulässigen Bereich gemäß der Perioden:

$$\mathcal{S}_t = \left\{ u \in \mathbb{R}^n : \sum_{i=1}^n u_i = \sigma_t, u \geq 0 \right\} \quad (31)$$

Wir können den zulässigen Bereich als kartesisches Produkt der eben definierten Mengen schreiben:

$$\mathcal{S} = \bigtimes_{t=1}^T \mathcal{S}_t \quad (32)$$

Das heißt, wir können für jede Periode unabhängig eine Gleichverteilung konstruieren und abschließend zu einer Gesamtlösung zusammenfassen.

#### 3.1. Konstruktion einer Gleichverteilung auf $\mathcal{S}_t$

Mit  $X_i$  bezeichnen wir die Zufallsvariable für  $x_{it}$  und mit  $f_i(\xi)$  bezeichnen wir die entsprechende Dichtefunktion. Wir gehen sequenziell vor. Beginnend mit der 1. Projektklasse ziehen wir der Reihe nach Zufallszahlen aus den entsprechenden Verteilungen:

1. Ziehe  $X_1 \sim f_1(\xi)$
2. Ziehe  $X_i \sim f_i(\xi) := f_{X_i|X_1, \dots, X_{i-1}}(\xi|\xi_1, \dots, \xi_{i-1})$  für  $i \in \{2, \dots, n-1\}$
3. Setze  $X_n = \sigma_t - \sum_{j=1}^{n-1} X_j$

Ab der 2. Projektklasse müssen wir dabei auf  $(X_1 = \xi_1, \dots, X_{i-1} = \xi_{i-1})$  bedingen. Bei der letzten Projektklasse wiederum haben wir keine Freiheiten mehr: Um die Zulässigkeitsbedingung  $\sum_{i=1}^n x_{it} = \sigma_t$  zu erfüllen, müssen wir die gesamte noch zur Verfügung stehende Restarbeitszeit in diese Projektklasse investieren.

Wie schauen  $f_1(\xi)$  und  $f_i(\xi)$  aus? Um diese Frage zu beantworten, definieren wir mit

$$\theta_1 = \sigma_t \quad (33)$$

$$\theta_i = \sigma_t - \sum_{j=1}^{i-1} \xi_j \quad \text{für } i \geq 2 \quad (34)$$

die zur Verfügung stehende Restarbeitszeit ab Projektklasse  $i$ .

**Lemma 13:** Um sequenziell eine Gleichverteilung zu generieren, müssen folgende Bedingungen erfüllt sein:

- $f_i(\xi) \propto (\theta_i - \xi)^{n-i-1}$  für  $i \in \{2, \dots, n-1\}$
- $f_1(\xi) \propto (\theta_1 - \xi)^{n-2} = (\sigma_t - \xi)^{n-2}$

*Beweis.* Zunächst nehmen wir die Menge  $\mathcal{S}_t$  etwas genauer unter die Lupe:

$$\begin{aligned} \mathcal{S}_t &= \left\{ \xi \in \mathbb{R}^n : \xi \geq 0, \sum_{j=1}^n \xi_j = \sigma_t \right\} \\ &= \left\{ \xi \in \mathbb{R}^n : \xi \geq 0, \sum_{j=1}^{n-1} \xi_j \leq \sigma_t, \xi_n = \sigma_t - \sum_{j=1}^{n-1} \xi_j \right\} \\ &= \left\{ \xi \in \mathbb{R}^n : \xi \geq 0, \xi_1 \leq \sigma_t, \xi_2 \leq \sigma_t - \xi_1, \dots, \xi_{n-1} \leq \sigma_t - \sum_{j=1}^{n-2} \xi_j, \xi_n = \sigma_t - \sum_{j=1}^{n-1} \xi_j \right\} \\ &= \{ \xi \in \mathbb{R}^n : \xi \geq 0, \xi_1 \leq \theta_1, \xi_2 \leq \theta_2, \dots, \xi_{n-1} \leq \theta_{n-1}, \xi_n = \theta_n \} \end{aligned}$$

Weiters definieren wir die folgenden Mengen:

$$\begin{aligned} \tilde{\mathcal{S}}_t &= \{ \xi \in \mathbb{R}^{n-1} : \xi \geq 0, \xi_1 \leq \theta_1, \xi_2 \leq \theta_2, \dots, \xi_{n-1} \leq \theta_{n-1} \} \\ \tilde{\mathcal{S}}_t^i &= \{ \xi = (\xi_i, \dots, \xi_{n-1})^T \in \mathbb{R}^{n-i} : \xi \geq 0, \xi_i \leq \theta_i, \xi_{i+1} \leq \theta_{i+1}, \dots, \xi_{n-1} \leq \theta_{n-1} \} \end{aligned}$$

Die Struktur von  $\mathcal{S}_t$  erlaubt es uns, in  $\mathcal{S}_t$  die Bedingung  $\xi_n = \theta_n$  wegzulassen und stattdessen eine Gleichverteilung auf  $\tilde{\mathcal{S}}_t$  zu betrachten. Gegeben  $(X_1 = \xi_1, \dots, X_{i-1} = \xi_{i-1})^T$  muss weiters der Vektor  $(X_i, \dots, X_{n-1})^T$  gleichverteilt auf  $\tilde{\mathcal{S}}_t^i$  sein.

Wenn  $f_{X_i, X_{i+1}, \dots, X_{n-1} | X_1, \dots, X_{i-1}}(\xi, \xi_{i+1}, \dots, \xi_{n-1} | \xi_1, \dots, \xi_{i-1}) \propto 1$  auf  $\tilde{\mathcal{S}}_t^i$  gilt, so ist eine Gleichverteilung auf der Menge  $\tilde{\mathcal{S}}_t^i$  gegeben. Wir sind an der Randdichte von  $X_i$  interessiert, weshalb wir  $X_{i+1}, \dots, X_{n-1}$  wegintegrieren müssen.

$$\begin{aligned} f_i(\xi) &= f_{X_i | X_1, \dots, X_{i-1}}(\xi | \xi_1, \dots, \xi_{i-1}) \\ &= \int_{\tilde{\mathcal{S}}_t^{i+1}} f_{X_i, X_{i+1}, \dots, X_{n-1} | X_1, \dots, X_{i-1}}(\xi, \xi_{i+1}, \dots, \xi_{n-1} | \xi_1, \dots, \xi_{i-1}) d\lambda^{n-i-1} \\ &\propto \int_0^{\theta_{i+1}} \int_0^{\theta_{i+2}} \dots \int_0^{\theta_{n-2}} \int_0^{\theta_{n-1}} 1 du_{n-1} du_{n-2} \dots du_{i+2} du_{i+1} \\ &= \int_0^{\theta_{i+1}} \int_0^{\theta_{i+2}} \dots \int_0^{\theta_{n-2}} \theta_{n-1} du_{n-2} \dots du_{i+2} du_{i+1} \\ &= \int_0^{\theta_{i+1}} \int_0^{\theta_{i+2}} \dots \underbrace{\int_0^{\theta_{n-2}} (\theta_{n-2} - u_{n-2}) du_{n-2} \dots du_{i+2} du_{i+1}}_{= \theta_{n-2}^2 - \frac{\theta_{n-2}^2}{2} \propto \theta_{n-2}^2} \end{aligned}$$

$$\begin{aligned} &\propto \dots \propto \int_0^{\theta_{i+1}} \theta_{i+2}^{n-i-2} du_{i+1} = \int_0^{\theta_{i+1}} (\theta_{i+1} - u_{i+1})^{n-i-2} du_{i+1} \\ &\propto -(\theta_{i+1} - u_{i+1})^{n-i-1} \Big|_0^{\theta_{i+1}} = \theta_{i+1}^{n-i-1} = (\theta_i - \xi)^{n-i-1} \end{aligned}$$

Analoge Rechnung für  $f_1(\xi)$  liefert  $f_1(\xi) \propto (\sigma_t - \xi)^{n-2}$ . □

Wegen der Stetigkeit von  $f_i(\cdot)$  können wir die **Inversionsmethode** (siehe zB [Kol08, Kapitel 8]) anwenden. Dazu müssen wir die Quantilsfunktion ausrechnen.

### Schritt 1: Integrationskonstante berechnen

Wir haben also die Dichte bis auf eine multiplikative Konstante bestimmt. Mit Hilfe des folgenden Ansatzes bestimmen wir jetzt diese Konstante:

$$f_i(\xi) = c \cdot (\theta_i - \xi)^{n-i-1} \text{ auf } [0, \theta_i]$$

Aus

$$1 = F(\theta_i) = c \cdot \int_0^{\theta_i} (\theta_i - \xi)^{n-i-1} d\xi = -c \cdot \frac{(\theta_i - \xi)^{n-i}}{n-i} \Big|_0^{\theta_i} = c \cdot \frac{\theta_i^{n-i}}{n-i}$$

folgt  $c = \frac{n-i}{\theta_i^{n-i}}$ . Also:

$$f_i(\xi) = \frac{n-i}{\theta_i^{n-i}} \cdot (\theta_i - \xi)^{n-i-1} \text{ auf } [0, \theta_i]$$

### Schritt 2: Verteilungsfunktion

$$F_i(\xi) = \int_0^\xi f_i(u) du = -\frac{n-i}{\theta_i^{n-i}} \cdot \frac{(\theta_i - u)^{n-i}}{n-i} \Big|_0^\xi = 1 - \frac{(\theta_i - \xi)^{n-i}}{\theta_i^{n-i}} \text{ auf } [0, \theta_i]$$

### Schritt 3: Quantilsfunktion

$$1 - \frac{(\theta_i - \xi)^{n-i}}{\theta_i^{n-i}} = p \Rightarrow Q(p) = \xi = \theta_i \cdot \left(1 - (1-p)^{\frac{1}{n-i}}\right)$$

## 3.2. Konstruktion einer Gleichverteilung auf $\mathcal{S}$

Nun haben wir das nötige Wissen, um für eine beliebige Periode eine Zufallslösung zu generieren. Die Ausdehnung auf mehrere Zeitpunkte ist eine Konsequenz der Annahme, dass die Perioden unabhängig sind. Somit können wir auf der nächsten Seite Algorithmus 1 niederschreiben.

Wir wenden in Abbildung 6 auf der nächsten Seite unsere Prozedur an. Unsere Rechnung geht wunderbar auf, gleichverteilter geht nicht! Wir werden uns vor allem in Kapitel 5 der Simplexgrafiken bedienen. Dort zeichnen wir als Bonus sogar Achsen ein!

---

**Algorithmus 1** Generiere gleichverteilte Zufallslösung

---

**Input:**  $T$  – die gewünschte Anzahl an Perioden

**Output:**  $x$  – eine zulässige gleichverteilte Lösung für  $T$  Perioden

```
1: Initialisiere  $x$  als  $n \times T$  Matrix
2: for  $t = 1$  to  $T$  do
3:    $\theta := \sigma_t$  // Restarbeitszeit für die aktuelle Periode
4:   for  $i = 1$  to  $n - 1$  do
5:      $p \leftarrow \text{Unif}(0, 1)$  // Gleichverteilte Zufallszahl in  $(0, 1]$ 
6:      $x_{it} \leftarrow \theta \cdot \left(1 - (1 - p)^{\frac{1}{n-i}}\right)$ 
7:      $\theta \leftarrow \theta - x_{it}$ 
8:   end for
9:    $x_{nt} \leftarrow \theta$ 
10: end for
11: return  $x$ 
```

---

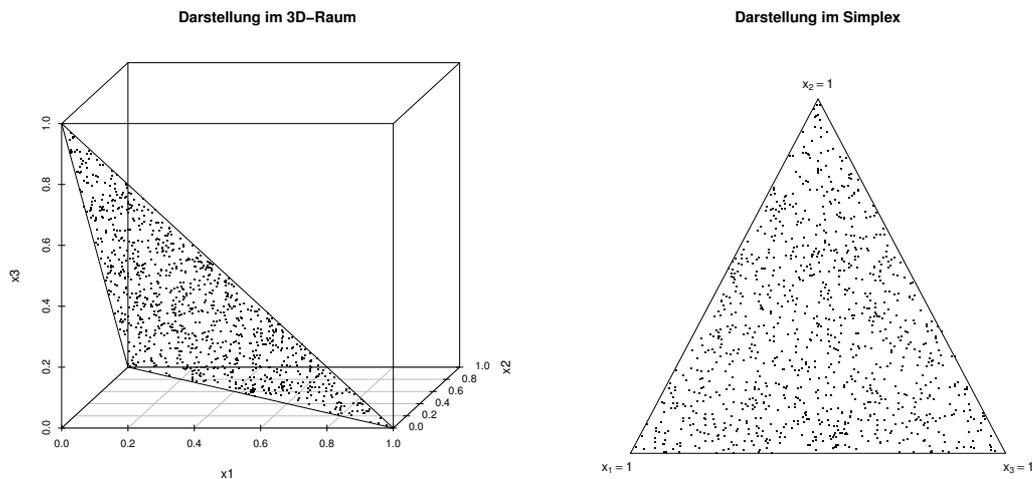


Abbildung 6: 1000 gleichverteilte Zufallslösungen für  $n = 3$  und  $T = 1$ . Wir simulieren mit  $\sigma = 1$ . **Links:** Dreieck im 3D-Raum. **Rechts:** Darstellung im Simplex. Diese Darstellung ist viel übersichtlicher. Die Ecke links unten entspricht der Lösung  $(1, 0, 0)^T$ , die obige  $(0, 1, 0)^T$  und jene rechts unten  $(0, 0, 1)^T$ .

## 4. Kurzfristige lokale Optimierung

In diesem Kapitel suchen wir nach Lösungen, welche den Gesamtertrag für eine gegebene Anzahl an Perioden lokal maximieren. Wir werden ein Gradientenverfahren herleiten, welches unsere Wünsche erfüllt. Dabei benötigen wir keine zusätzlichen Annahmen.

In Abschnitt 4.1 berechnen wir den Gradienten, also die Richtung des steilsten Anstiegs. Schon im darauf folgenden Abschnitt stellen wir fest, dass diese Richtung in der Regel leider nicht zulässig ist. Wie wir aus dem Gradienten eine zulässige verbessernde Richtung basteln können, erläutern wir in Abschnitt 4.3.

In Abschnitt 4.4 diskutieren wir notwendige Bedingungen für lokale Optima und in 4.5 überlegen wir uns, wie wir Sattelpunkten entkommen können. Im letzten Abschnitt 4.7 schreiben wir einen Iterated Local Search Algorithmus auf.

### 4.1. Gradient

Um eine beliebige zulässige Lösung lokal zu optimieren, verwenden wir im Prinzip die Methode des steilsten Anstiegs. Wir fangen mit dem Grundbaustein an, dem Gradienten. Werfen wir nochmals einen Blick auf die Zielfunktion:

$$G(x) = \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} \rightarrow \max! \quad (35)$$

wobei

$$z_{rt} = z_{rt-1} - \beta_r + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \eta_{it-1} x_{it-1} \quad \text{für } t \geq 2 \quad (36)$$

$$\gamma_{rt} = \varphi(z_{rt}) = \frac{1}{1 + a \cdot \exp(-b \cdot z_{rt})} \quad (37)$$

$$\eta_{it} = \left( \sum_{\rho=1}^R \frac{d_{i\rho}}{\gamma_{\rho t}} \right)^{-1} \quad (38)$$

Wir berechnen die partiellen Ableitungen. Dabei verwenden wir folgende Notation:

$$\dot{G}_x^{js} = \frac{\partial G(x)}{\partial x_{js}} \quad \dot{\eta}_{it}^{js} = \frac{\partial \eta_{it}}{\partial x_{js}} \quad \dot{\gamma}_{rt}^{js} = \frac{\partial \gamma_{rt}^{-1}}{\partial x_{js}} \quad \dot{z}_{rt}^{js} = \frac{\partial z_{rt}}{\partial x_{js}} \quad (39)$$

Der Gradient hat in unserem Problem die Gestalt einer Matrix:

$$\dot{G}_x = \nabla G(x) := \begin{bmatrix} \frac{\partial G(x)}{\partial x_{11}} & \frac{\partial G(x)}{\partial x_{12}} & \cdots & \frac{\partial G(x)}{\partial x_{1T}} \\ \frac{\partial G(x)}{\partial x_{21}} & \frac{\partial G(x)}{\partial x_{22}} & \cdots & \frac{\partial G(x)}{\partial x_{2T}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G(x)}{\partial x_{n1}} & \frac{\partial G(x)}{\partial x_{n2}} & \cdots & \frac{\partial G(x)}{\partial x_{nT}} \end{bmatrix} = \begin{bmatrix} \dot{G}_x^{11} & \dot{G}_x^{12} & \cdots & \dot{G}_x^{1T} \\ \dot{G}_x^{21} & \dot{G}_x^{22} & \cdots & \dot{G}_x^{2T} \\ \vdots & \vdots & \ddots & \vdots \\ \dot{G}_x^{n1} & \dot{G}_x^{n2} & \cdots & \dot{G}_x^{nT} \end{bmatrix} \quad (40)$$

Wir fangen mit den partiellen Ableitungen der Zielfunktion an.

$$\dot{G}_x^{js} = \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \frac{\partial}{\partial x_{js}} (\eta_{it} x_{it}) \quad (41)$$

Nun lösen wir die Ableitungsausdrücke der Reihe nach auf.

$$\frac{\partial}{\partial x_{js}} (\eta_{it} x_{it}) = \begin{cases} 0 & \text{für } s > t \\ 0 & \text{für } s = t, j \neq i \\ \eta_{jt} & \text{für } s = t, j = i \\ \dot{\eta}_{it}^{js} x_{it} & \text{für } s < t \end{cases} \quad (42)$$

$$\dot{\eta}_{it}^{js} = \begin{cases} 0 & \text{für } s \geq t \\ -\eta_{it}^2 \sum_{\rho=1}^R d_{i\rho} \dot{\gamma}_{\rho t}^{js} & \text{für } s < t \end{cases} \quad (43)$$

$$\dot{\gamma}_{rt}^{js} = \begin{cases} 0 & \text{für } s \geq t \\ -a \cdot b \cdot e^{-b \cdot z_{rt}} \dot{z}_{rt}^{js} & \text{für } s < t \end{cases} \quad (44)$$

Da  $\gamma_{rt}$  und damit folglich auch  $\eta_{it}$  Funktionen in  $x_1, \dots, x_{t-1}$  sind, fallen alle Ableitungen in (43) und (44) weg, wo  $s \geq t$  ist. Der Fall  $s < t$  ist nur noch Formsache. In (42) müssen wir für  $s = t$  noch eine Fallunterscheidung treffen.

Wir können also die Berechnung des Gradienten auf die Ausdrücke  $\dot{z}_{rt}^{js}$  zurückführen.

#### Ableitung von $z_{rt}$

*Fall:  $s \geq t$ :*

Da  $z_{rt}$  eine Funktion in  $x_1, \dots, x_{t-1}$  ist, fallen alle Ableitungen weg, das heißt es gilt:

$$\dot{z}_{rt}^{js} = 0 \quad \text{für } s \geq t$$

*Fall:  $s = t - 1$ :*

Hier fallen analog zu oben einige Ausdrücke weg.

$$\begin{aligned}
\dot{z}_{rt}^{js} &= \underbrace{\dot{z}_{rt-1}^{js}}_{=0} + \alpha_r \sum_{i=1}^n d_{ir} \underbrace{\dot{\gamma}_{rt-1}^{js}}_{=0} \eta_{it-1} x_{it-1} + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \cdot \underbrace{\frac{\partial}{\partial x_{js}} (\eta_{it-1} x_{it-1})}_{=0 \text{ für } i \neq j} \\
&= \alpha_r \frac{d_{jr}}{\gamma_{rt-1}} \cdot \frac{\partial}{\partial x_{js}} (\eta_{jt-1} x_{jt-1}) \\
&= \alpha_r \frac{d_{jr}}{\gamma_{rt-1}} \eta_{jt-1} \quad \text{für } s = t - 1
\end{aligned}$$

Fall:  $s < t - 1$ :

Hier fällt zwar nichts weg, allerdings können wir die Ausdrücke der Vorperiode verwenden:

$$\begin{aligned}
\dot{z}_{rt}^{js} &= \dot{z}_{rt-1}^{js} + \alpha_r \sum_{i=1}^n d_{ir} \dot{\gamma}_{rt-1}^{js} \eta_{it-1} x_{it-1} + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \cdot \frac{\partial}{\partial x_{js}} (\eta_{it-1} x_{it-1}) \\
&= \dot{z}_{rt-1}^{js} + \alpha_r \sum_{i=1}^n d_{ir} \dot{\gamma}_{rt-1}^{js} \eta_{it-1} x_{it-1} + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \dot{\eta}_{it-1}^{js} x_{it-1} \quad \text{für } s < t - 1
\end{aligned}$$

Die Ableitungen von  $z_{rt}$  auf einen Blick:

$$\dot{z}_{rt}^{js} = \begin{cases} 0 & \text{für } s \geq t \\ \alpha_r \frac{d_{jr}}{\gamma_{rt-1}} \eta_{jt-1} & \text{für } s = t - 1 \\ \dot{z}_{rt-1}^{js} + \alpha_r \sum_{i=1}^n d_{ir} \dot{\gamma}_{rt-1}^{js} \eta_{it-1} x_{it-1} + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \dot{\eta}_{it-1}^{js} x_{it-1} & \text{für } s < t - 1 \end{cases}$$

Mit dem Wissen aus (42) können wir die Ableitung der Zielfunktion spezifizieren:

$$\begin{aligned}
\dot{G}_x^{js} &= \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \frac{\partial}{\partial x_{js}} (\eta_{it} x_{it}) \\
&= \sum_{t=s}^T \delta^{-t} \sum_{i=1}^n w_{it} \frac{\partial}{\partial x_{js}} (\eta_{it} x_{it}) \\
&= \delta^{-s} w_{js} \eta_{js} + \sum_{t=s+1}^T \delta^{-t} \sum_{i=1}^n w_{it} \dot{\eta}_{it}^{js} x_{it} \tag{45}
\end{aligned}$$

Mit der richtigen Berechnungsreihenfolge können die Ableitungen  $\dot{G}_x^{js}$  berechnet werden. Algorithmus 2 auf der nächsten Seite zeigt, wie.

**Bemerkung 14:** Die in Algorithmus 2 dargestellte Variante ist nicht die arbeitsspeichereffizienteste, dafür glänzt sie mit ihrer Übersichtlichkeit.

---

**Algorithmus 2** Berechne den Gradienten

---

**Input:**  $x$  – eine zulässige Lösung

**Output:**  $\dot{G}_x$  – der Gradient ausgewertet bei  $x$

```
1:  $\dot{z}_{r1}^{js} \leftarrow 0 \quad \forall r \in \mathcal{R}, j \in \mathcal{N}, s \in \mathcal{T}$ 
2:  $\dot{\gamma}_{r1}^{js} \leftarrow 0 \quad \forall r \in \mathcal{R}, j \in \mathcal{N}, s \in \mathcal{T}$ 
3:  $\dot{\eta}_{i1}^{js} \leftarrow 0 \quad \forall i, j \in \mathcal{N}, s \in \mathcal{T}$ 
4:  $\dot{G}_x^{js} \leftarrow \delta^{-s} w_{js} \eta_{js} \quad \forall j \in \mathcal{N}, s \in \mathcal{T}$  // Fall:  $s = t$ , wird in Schleife nicht abgedeckt
5: for  $t = 2$  to  $T$  do
6:   for  $s = 1$  to  $t - 1$  do
7:     if  $s = t - 1$  then
8:        $\dot{z}_{rt}^{js} \leftarrow \alpha_r \frac{d_{jr}}{\gamma_{rt-1}} \eta_{jt-1} \quad \forall r, j$ 
9:     else
10:       $\dot{z}_{rt}^{js} \leftarrow \dot{z}_{rt-1}^{js} + \alpha_r \sum_{i=1}^n d_{ir} \dot{\gamma}_{rt-1}^{js} \eta_{it-1} x_{it-1} + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt-1}} \dot{\eta}_{it-1}^{js} x_{it-1} \quad \forall r, j$ 
11:    end if
12:     $\dot{\gamma}_{rt}^{js} \leftarrow -a \cdot b \cdot e^{-b \cdot z_{rt}} \dot{z}_{rt}^{js} \quad \forall r, j$ 
13:     $\dot{\eta}_{it}^{js} \leftarrow -\eta_{it}^2 \sum_{\rho=1}^R d_{i\rho} \dot{\gamma}_{\rho t}^{js} \quad \forall i, j$ 
14:     $\dot{G}_x^{js} \leftarrow \dot{G}_x^{js} + \delta^{-t} \sum_{i=1}^n w_{it} \dot{\eta}_{it}^{js} x_{it} \quad \forall j$ 
15:  end for
16: end for
17: return  $\dot{G}_x$ 
```

---

## 4.2. Zulässige und verbessernde Richtungen

Wir befassen uns im Folgenden mit der Frage, welche Bedingungen für eine Richtung gelten müssen, damit sie zulässig ist.

**Definition 15:** Sei  $x$  eine zulässige Lösung. Die Matrix

$$\Delta = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \dots & \Delta_{1T} \\ \Delta_{21} & \Delta_{22} & \dots & \Delta_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{n1} & \Delta_{n2} & \dots & \Delta_{nT} \end{bmatrix} \neq 0$$

heißt **zulässige Richtung** in  $x$ , falls  $x + \epsilon \cdot \Delta$  zulässig ist für alle hinreichend kleinen  $\epsilon > 0$ , also falls für alle hinreichend kleinen  $\epsilon > 0$  folgende Bedingungen erfüllt sind:

$$\sum_{i=1}^n (x_{it} + \epsilon \cdot \Delta_{it}) = \sigma_t \quad \forall t \in \mathcal{T} \quad (46)$$

$$x_{it} + \epsilon \cdot \Delta_{it} \geq 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (47)$$

Dies ist eine allgemeine Definition, die wir im folgenden Satz auf unser Optimierungsproblem maßschneidern werden.

**Satz 16:** Sei  $x$  eine zulässige Lösung. Die Matrix

$$\Delta = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \dots & \Delta_{1T} \\ \Delta_{21} & \Delta_{22} & \dots & \Delta_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{n1} & \Delta_{n2} & \dots & \Delta_{nT} \end{bmatrix} \neq 0$$

ist eine **zulässige Richtung** in  $x$ , falls folgende Bedingungen erfüllt sind:

$$\sum_{i=1}^n \Delta_{it} = 0 \quad \forall t \in \mathcal{T} \quad (48)$$

$$\Delta_{it} \geq 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} : x_{it} = 0 \quad (49)$$

*Beweis.* Wenn (48) erfüllt ist, so gilt für alle  $\epsilon > 0$

$$\sum_{i=1}^n (x_{it} + \epsilon \cdot \Delta_{it}) = \underbrace{\sum_{i=1}^n x_{it}}_{=\sigma_t} + \epsilon \cdot \underbrace{\sum_{i=1}^n \Delta_{it}}_{=0} = \sigma_t \quad \forall t \in \mathcal{T}$$

womit (46) gezeigt wäre. Um (47) zu zeigen, unterscheiden wir zwei Fälle.

*Fall:*  $x_{it} > 0$

Wir stellen fest, dass  $x_{it} + \epsilon \cdot \Delta_{it} \geq 0$  für alle hinreichend kleinen  $\epsilon > 0$  erfüllt ist.

*Fall:*  $x_{it} = 0$

Aus Bedingung (49) folgt für beliebiges  $\epsilon > 0$ :

$$x_{it} + \epsilon \cdot \Delta_{it} \geq 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} : x_{it} = 0$$

□

Richtungen sollen nicht nur zulässig sein, sie sollen uns auch dabei helfen, den Zielfunktionswert zu steigern.

**Definition 17:** Sei  $x$  eine zulässige Lösung. Die Matrix

$$\Delta = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \dots & \Delta_{1T} \\ \Delta_{21} & \Delta_{22} & \dots & \Delta_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{n1} & \Delta_{n2} & \dots & \Delta_{nT} \end{bmatrix} \neq 0$$

heißt **zulässige Verbesserungsrichtung** in  $x$  für  $G(\cdot)$ , falls sie zulässig ist und darüber hinaus noch folgende Bedingung erfüllt:

$$\sum_{s=1}^T \sum_{j=1}^n \dot{G}_x^{js} \cdot \Delta_{js} > 0 \quad (50)$$

### 4.3. Korrigierter Gradient

Da in endlicher Zeit stets  $\gamma_{rt} > 0$  gilt und wir für alle Erträge  $w_{it} > 0$  annehmen, ist der Gradient  $\nabla G(x)$  komponentenweise positiv. Wir würden am liebsten unendlich viel Arbeitszeit in alle Projektklassen stecken. Ausgehend von einer zulässigen Lösung  $x$  wäre  $x + \epsilon \cdot \nabla G(x)$  unzulässig für jedes  $\epsilon > 0$ , da die Gleichheitsbedingung (47) verletzt wäre.

Der Gradient ist also in der Regel keine zulässige Richtung, folglich müssen wir korrigierend eingreifen. Wir werden uns hier eine einfache Heuristik ansehen, welche aus dem Gradienten eine zulässige Verbesserungsrichtung bastelt.

Ausgehend von einer zulässigen Lösung  $x$  gibt  $\dot{G}_x^{js}$  an, wie stark die Zielfunktion  $G$  steigen würde, wenn wir  $x_{js}$  ein klein wenig erhöhen. Je größer dieser Wert, desto lukrativer ist eine solche Erhöhung.

Nun greifen wir uns innerhalb der selben Periode  $s$  eine zweite Projektklasse  $i$  heraus und vergleichen beide Ableitungen miteinander. Gilt  $\dot{G}_x^{js} > \dot{G}_x^{is}$ , dann steigt die Zielfunktion, wenn wir in Periode  $s$  etwas Arbeitszeit von Projektklasse  $i$  nach  $j$  verschieben.

Diese Idee werden wir nun verallgemeinern. Wir bezeichnen mit  $\ddot{G}_x$  den **korrigierten Gradienten** und weisen ihm zunächst den normalen Gradienten zu, also

$$\ddot{G}_x \leftarrow \dot{G}_x \quad (51)$$

Im Folgenden sei eine beliebige Periode  $s \in \mathcal{T}$  gegeben. Wir führen für die  $s$ . Spalte von  $\ddot{G}_x$  eine **Mittelwertbereinigung** durch:

$$\ddot{G}_x^{js} \leftarrow \ddot{G}_x^{js} - \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \ddot{G}_x^{is} \quad \forall j \in \mathcal{N} \quad (52)$$

Wir erhalten dadurch die Information, welche Projektklassen in Periode  $s$  bei einer Erhöhung der Arbeitszeit einen unterdurchschnittlichen und welche einen überdurchschnittlichen Zielfunktionszuwachs zur Folge hätten. Mit dieser Korrektur sichten wir Arbeitszeit auf erfolgversprechendere Projektklassen um.

Es ist klar, dass dadurch die Gleichheitsbedingung (48) erfüllt ist. Aber unter Umständen ist Bedingung (49) verletzt, nämlich genau dann, wenn  $x_{js} = 0$  und gleichzeitig  $\ddot{G}_x^{js} < 0$  gilt. Hier würde  $x_{js}$  negativ werden, was illegal ist.

Um dem entgegenzuwirken, setzen wir in diesem Fall  $\ddot{G}_x^{js} = 0$  (**Nullsetzen** von Projektklasse  $i$ ) und **streichen** diese Projektklasse aus unseren weiteren Überlegungen. Dadurch wird aber in der Regel die Gleichheitsbedingung (48) wieder verletzt...

Wir haben also die folgenden beiden Werkzeuge:

- **Mittelwertbereinigung** der  $s$ . Spalte von  $\ddot{G}_x$ , um die Gleichheitsbedingung (48) sicherzustellen.
- **Null-Setzen** und **Streichen** von Projektklassen, für die  $x_{js} = 0$  und gleichzeitig  $\ddot{G}_x^{js} < 0$  gilt, um Bedingung (49) zu gewährleisten.

Beide Werkzeuge werden so lange alternierend angewendet, bis beide Bedingungen erfüllt sind. Diese Prozedur, die für alle Perioden  $s \in \mathcal{T}$  durchgeführt wird, ist in Algorithmus 3 niedergeschrieben.

---

**Algorithmus 3** Berechne den korrigierten Gradienten

---

**Input:**  $x$  – eine zulässige Lösung

**Output:**  $\ddot{G}_x$  – der korrigierte Gradient ausgewertet bei  $x$

```

1:  $\dot{G}_x \leftarrow$  Berechne den Gradienten gemäß Algorithmus 2
2: for  $s = 1$  to  $T$  do
3:    $\ddot{N}_s \leftarrow \{1, \dots, n\}$  // Menge aller nichtgestrichenen Projektklassen in Periode  $s$ 
4:   repeat
5:      $\theta \leftarrow 0$ 
6:     for all  $j \in \ddot{N}_s$  do
7:       if  $\ddot{G}_x^{js} < 0 \wedge x_{js} = 0$  then
8:          $\ddot{G}_x^{js} \leftarrow 0$  // Null-Setzen der Klasse  $j$ 
9:          $\ddot{N}_s \leftarrow \ddot{N}_s / \{j\}$  // Streichung der Klasse  $j$ 
10:      else
11:         $\theta \leftarrow \theta + \ddot{G}_x^{js}$ 
12:      end if
13:    end for
14:    for all  $j \in \ddot{N}_s$  do
15:       $\ddot{G}_x^{js} \leftarrow \ddot{G}_x^{js} - \frac{\theta}{|\ddot{N}_s|}$  // Mittelwertsbereinigung
16:    end for
17:  until  $\theta = 0$ 
18: end for
19: return  $\ddot{G}_x$ 

```

---

**Beispiel 18 (1. Gradientenbeispiel):** Ein Beispiel mit  $n = 4$ ,  $T = 1$  und  $\sigma = 1$ .

$$x = \begin{bmatrix} 0.2 \\ 0.8 \\ 0 \\ 0 \end{bmatrix} \quad \dot{G}_x = \begin{bmatrix} 7 \\ 15 \\ 8 \\ 2 \end{bmatrix} \quad \mathcal{N} = \{1, 2, 3, 4\}$$

Iteration 1:

$$\ddot{N} = \{1, 2, 3, 4\} \\ \theta = 32 \text{ und } \theta/|\ddot{N}| = 8$$

$$\ddot{G}_x = \begin{pmatrix} 7-8 \\ 15-8 \\ 8-8 \\ 2-8 \end{pmatrix} = \begin{pmatrix} -1 \\ 7 \\ 0 \\ -6 \end{pmatrix}$$

Iteration 2:

$$\ddot{N} = \{1, 2, 3\} \\ \theta = 6 \text{ und } \theta/|\ddot{N}| = 2$$

$$\ddot{G}_x = \begin{pmatrix} -1-2 \\ 7-2 \\ 0-2 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ 5 \\ -2 \\ 0 \end{pmatrix}$$

Iteration 3:

$$\ddot{N} = \{1, 2\} \\ \theta = 2 \text{ und } \theta/|\ddot{N}| = 1$$

$$\ddot{G}_x = \begin{pmatrix} -3-1 \\ 5-1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -4 \\ 4 \\ 0 \\ 0 \end{pmatrix}$$

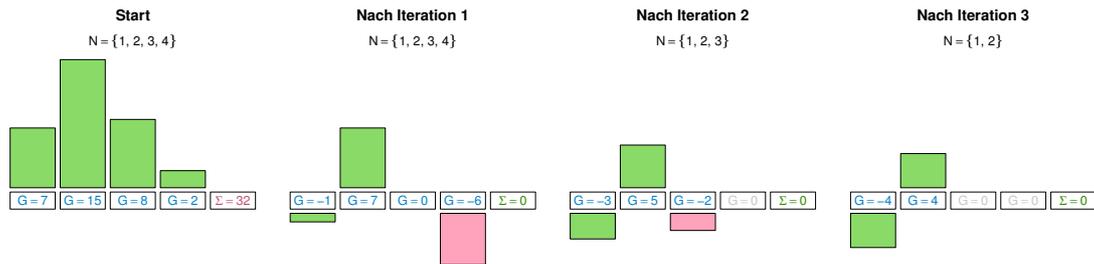


Abbildung 7: Entwicklung des korrigierten Gradienten im 1. Gradientenbeispiel. Verletzungen der Zulässigkeitsbedingungen sind rot markiert.

Am Anfang ist die Summenbedingung (48) nicht erfüllt. Nach der Mittelwertsbereinigung in Iteration 1 ist  $\ddot{G}_x^{41} < 0$ . Wir dürfen aber  $x_{41}$  nicht senken, da  $x_{41}$  sonst negativ wäre. Also streichen wir Klasse 4 in Iteration 2 und setzen  $\ddot{G}_x^{41} = 0$ . Das selbe Problem tritt bei Projektklasse 3 auf, das wir in Iteration 3 auf die gleiche Art und Weise lösen. Nach der Mittelwertsbereinigung ist die Richtung zulässig.

**Beispiel 19 (2. Gradientenbeispiel):** Das gleiche Beispiel leicht abgewandelt.

$$x = \begin{bmatrix} 0.2 \\ 0.8 \\ 0 \\ 0 \end{bmatrix} \quad \dot{G}_x = \begin{bmatrix} 6 \\ 15 \\ 12 \\ 3 \end{bmatrix} \quad \mathcal{N} = \{1, 2, 3, 4\}$$

Iteration 1:

$$\tilde{\mathcal{N}} = \{1, 2, 3, 4\} \\ \theta = 36 \text{ und } \theta/|\tilde{\mathcal{N}}| = 9$$

Iteration 2:

$$\tilde{\mathcal{N}} = \{1, 2, 3\} \\ \theta = 6 \text{ und } \theta/|\tilde{\mathcal{N}}| = 2$$

$$\ddot{G}_x = \begin{pmatrix} 6 - 9 \\ 15 - 9 \\ 12 - 9 \\ 3 - 9 \end{pmatrix} = \begin{pmatrix} -3 \\ 6 \\ 3 \\ -6 \end{pmatrix} \quad \ddot{G}_x = \begin{pmatrix} -3 - 2 \\ 6 - 2 \\ 3 - 2 \\ 0 \end{pmatrix} = \begin{pmatrix} -5 \\ 4 \\ 1 \\ 0 \end{pmatrix}$$

Im Gegensatz zum vorherigen Beispiel wird etwas Arbeitszeit auch auf Projektklasse 3 umgeschauvelt. Abbildung 8 auf der nächsten Seite zeigt die Situation.

**Satz 20:** Algorithmus 3 liefert uns, sofern  $\ddot{G}_x \neq 0$ , eine zulässige Verbesserungsrichtung.

*Beweis.* Da die Perioden unabhängig sind, genügt es, sich eine beliebige Periode  $s \in \mathcal{T}$  anzusehen. Für diese Periode nehmen wir die innere Schleife von Algorithmus 3 (Zeilen 4 bis 17) genauer unter die Lupe.

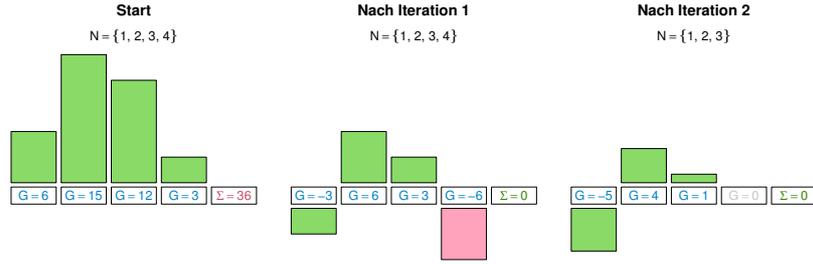


Abbildung 8: Entwicklung des korrigierten Gradienten im 2. Gradientenbeispiel. Verletzungen der Zulässigkeitsbedingungen sind rot markiert.

In Anlehnung an Satz 16 auf Seite 43 müssen wir für Periode  $s$  folgendes nachrechnen, um zu zeigen, dass  $\check{G}_x$  eine **zulässige Richtung** ist:

$$\sum_{j=1}^n \check{G}_x^{js} = 0 \quad (53)$$

$$\check{G}_x^{js} \geq 0 \quad \forall j \in \mathcal{N} : x_{js} = 0 \quad (54)$$

In jeder Iteration werden zunächst in den Zeilen 6 bis 13 alle negativen Einträge der  $s$ . Spalte von  $\check{G}_x$ , wo  $x_{js} = 0$  gilt, auf Null gesetzt und aus den weiteren Überlegungen entfernt. Nach Zeile 13 ist also Bedingung (54) stets erfüllt.

Jetzt gibt es zwei Möglichkeiten:

*Fall:  $\theta = 0$*

Hier ist die Summenbedingung (53) bereits erfüllt, eine Mittelwertsbereinigung ist nicht nötig. Dadurch bleibt Bedingung (54) erfüllt und wir können die Schleife getrost verlassen.

*Fall:  $\theta \neq 0$*

In diesem Fall ist eine Mittelwertsbereinigung notwendig, um die Summenbedingung (53) zu erfüllen. Möglicherweise wird dadurch aber (wieder) Bedingung (54) verletzt. Also wiederholen wir die Schleife.

Diese Schleife wird aber höchstens  $n$  Mal durchlaufen, da die Menge  $\check{\mathcal{N}}_s$  ab Durchgang 2 echt kleiner wird, sofern der zweite Fall eintritt.

Unterm Strich ist also der korrigierte Gradient eine zulässige Richtung. Wenden wir uns nun der **Verbesserungsrichtung** zu.

Sei  $s \in \mathcal{T}$ . Für  $j \in \check{\mathcal{N}}_s$  gilt am Ende des Algorithmus

$$\check{G}_x^{js} = \dot{G}_x^{js} - \frac{1}{|\check{\mathcal{N}}_s|} \sum_{i \in \check{\mathcal{N}}_s} \dot{G}_x^{is} \quad (55)$$

Jetzt:

$$\begin{aligned} \sum_{j \in \check{\mathcal{N}}_s} \dot{G}_x^{js} \cdot \ddot{G}_x^{js} &= \sum_{j \in \check{\mathcal{N}}_s} \dot{G}_x^{js} \cdot \dot{G}_x^{js} - \sum_{j \in \check{\mathcal{N}}_s} \dot{G}_x^{js} \cdot \frac{1}{|\check{\mathcal{N}}_s|} \sum_{i \in \check{\mathcal{N}}_s} \dot{G}_x^{is} \\ &= \sum_{j \in \check{\mathcal{N}}_s} \left( \dot{G}_x^{js} \right)^2 - \frac{1}{|\check{\mathcal{N}}_s|} \left( \sum_{j \in \check{\mathcal{N}}_s} \dot{G}_x^{js} \right)^2 = \sum_{j \in \check{\mathcal{N}}_s} \left( \dot{G}_x^{js} - \frac{1}{|\check{\mathcal{N}}_s|} \sum_{i \in \check{\mathcal{N}}_s} \dot{G}_x^{is} \right)^2 \geq 0 \end{aligned}$$

Letzte Gleichung ist eine simple Anwendung des Verschiebungssatzes. Nach einem Blick auf (55) erkennen wir, dass letzte Ungleichung nur dann mit Gleichheit erfüllt ist, wenn:

$$\ddot{G}_x^{js} = 0 \quad \forall j \in \check{\mathcal{N}}_s$$

Abschließend:

$$\sum_{j \in \mathcal{N}} \dot{G}_x^{js} \cdot \ddot{G}_x^{js} = \sum_{j \notin \check{\mathcal{N}}_s} \dot{G}_x^{js} \cdot \underbrace{\ddot{G}_x^{js}}_{=0} + \sum_{j \in \check{\mathcal{N}}_s} \dot{G}_x^{js} \cdot \ddot{G}_x^{js} \geq 0$$

Diese Aussage gilt für alle  $s \in \mathcal{T}$ , folglich gilt:

$$\sum_{s \in \mathcal{T}} \sum_{j \in \mathcal{N}} \dot{G}_x^{js} \cdot \ddot{G}_x^{js} \geq 0$$

Gleichheit ist aber, wie wir gesehen haben, nur dann möglich, falls  $\ddot{G}_x = 0$ . Das heißt, falls  $\ddot{G}_x \neq 0$ , so ist diese Richtung verbessernd. Damit ist dieser Satz bewiesen.  $\square$

Die Nullmatrix hat eine besondere Bedeutung. Welche genau sie hat, sehen wir im folgenden Unterabschnitt.

#### 4.4. Notwendige Bedingungen für lokale Optima

Zunächst schreiben wir uns notwendige Bedingungen für den allgemeinen Fall auf und werden diese dann auf unser Optimierungsproblem maßschneidern. Für den allgemeinen Fall betrachten wir das folgende Optimierungsproblem:

$$(\hat{P}) \begin{cases} f(x) \rightarrow \min! \\ g_i(x) \leq 0 \quad i = 1, \dots, m_1 \\ h_j(x) = 0 \quad j = 1, \dots, m_2 \end{cases}$$

Dabei ist  $x \in \mathbb{R}^n$ ,  $f : \mathbb{R}^n \mapsto \mathbb{R}$ , alle  $g_i : \mathbb{R}^n \mapsto \mathbb{R}$  und alle  $h_j : \mathbb{R}^n \mapsto \mathbb{R}$ . Den zulässigen Bereich beschreibt die Menge

$$\mathcal{M} = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \quad \forall i = 1, \dots, m_1, h_j(x) = 0 \quad \forall j = 1, \dots, m_2\}$$

Eine Ungleichungsnebenbedingung heißt aktiv in  $x \in \mathcal{M}$ , falls  $g_i(x) = 0$ . Wir fassen die Menge aller aktiven Ungleichungsbedingungen zusammen:

$$I(x) = \{i \in \{1, \dots, m_1\} : g_i(x) = 0\}$$

**Satz 21 (Karush/Kuhn/Tucker; KKT):** Sei  $x$  eine lokale Minimalstelle von  $(\hat{P})$  auf  $\mathcal{M}$ . Seien  $f$ , alle  $g_i$  und  $h_j$  stetig differenzierbar auf  $\mathcal{M}$ . Seien weiters die Gradienten der in  $x$  aktiven Ungleichheitsnebenbedingungen und die Gradienten aller Gleichheitsbedingungen linear unabhängig. Dann gilt: Es existieren Lagrange-Multiplikatoren  $\lambda_i \geq 0$  und  $\mu_j \in \mathbb{R}$ , sodass:

$$\nabla f(x) + \sum_{i=1}^{m_1} \lambda_i \nabla g_i(x) + \sum_{j=1}^{m_2} \mu_j \nabla h_j(x) = 0 \quad (56)$$

$$\lambda_i g_i(x) = 0 \quad \forall i = 1, \dots, m_1 \quad (57)$$

*Beweis.* Siehe zB [Bom98] oder [Möh06]. □

Nun zu unserem Optimierungsproblem. Hierbei greifen wir nur die notwendigen Teile heraus und formulieren sie um:

$$(P) \begin{cases} G(x) \rightarrow \max! \\ g_{it}(x) = -x_{it} \leq 0 \quad i \in \mathcal{N}, t \in \mathcal{T} \\ h_t(x) = \sum_{i=1}^n x_{it} - \sigma_t = 0 \quad t \in \mathcal{T} \end{cases}$$

Dabei ist  $x \in \mathbb{R}^{n \times T}$ . Die Funktionen  $G(\cdot)$ , alle  $g_{it}(\cdot)$  und  $h_t(\cdot)$  sind stetig differenzierbar auf der zulässigen Menge

$$\mathcal{S} = \{x \in \mathbb{R}^{n \times T} : g_{it}(x) \leq 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, h_t(x) = 0 \quad \forall t \in \mathcal{T}\} \quad (58)$$

Wie oben fassen wir alle aktiven Ungleichheitsbedingungen zusammen, hier allerdings separiert nach den Perioden:

$$I_t(x) = \{i \in \mathcal{N} : g_{it}(x) = 0\} = \{i \in \mathcal{N} : x_{it} = 0\} \quad \forall t \in \mathcal{T} \quad (59)$$

Die Gradienten  $\nabla g_{it}(x)$  und  $\nabla h_t(x)$  haben folgendes Aussehen:

$$\nabla g_{it}(x) = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad \nabla h_t(x) = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \quad (60)$$

Der Gradient  $\nabla g_{it}(x)$  ist die Nullmatrix, wobei in der  $i$ . Zeile und  $t$ . Spalte  $-1$  steht und  $\nabla h_t(x)$  ist ebenfalls die Nullmatrix, wobei die  $t$ . Spalte aus lauter Einsen besteht.

**Lemma 22:** Sei  $x$  eine beliebige zulässige Lösung. Sei

$$\mathcal{G} = \bigcup_{t=1}^T \{\nabla g_{it}(x) : i \in I_t(x)\} \cup \{\nabla h_t(x)\} \quad (61)$$

die Menge aller Gradienten, welche in  $x$  aktiv sind. Dann gilt: Die Gradienten aus  $\mathcal{G}$  sind linear unabhängig.

*Beweis.* Zunächst beobachten wir mit Hilfe von (60), dass die Gradienten periodenweise linear unabhängig sind. Das heißt: Es genügt zu zeigen, dass für beliebiges  $t \in \mathcal{T}$  die Gradienten in der Menge

$$\{\nabla g_{it}(x) : i \in I_t(x)\} \cup \{\nabla h_t(x)\}$$

linear unabhängig sind. Offensichtlich kann diese Menge aber nur dann linear abhängig sein, falls  $I_t(x) = \mathcal{N} = \{1, \dots, n\}$ , also wenn alle Ungleichungsnebenbedingungen aktiv sind. Das kann allerdings nicht passieren, da in diesem Falle  $\sum_{i=1}^n x_{it} = 0 \neq \sigma_t > 0$  gelten würde, was wiederum die Unzulässigkeit von  $x$  zur Folge hätte – ein Widerspruch! Also muss es immer mindestens eine Nebenbedingung geben, die nicht aktiv ist, womit das Lemma bewiesen ist.  $\square$

Nun werden wir uns notwendige Bedingungen herleiten.

**Satz 23:** Sei  $x$  ein lokales Maximum von  $(P)$ . Dann gilt für jedes  $t \in \mathcal{T}$ :

$$\dot{G}_x^{it} = \dot{G}_x^{jt} \quad \forall i \notin I_t(x), j \in I_t(x) \quad (62)$$

$$\dot{G}_x^{it} \geq \dot{G}_x^{jt} \quad \forall i \notin I_t(x), j \in I_t(x) \quad (63)$$

*Beweis.* Wir haben in Lemma 22 bereits gezeigt, dass die Gradienten der aktiven Nebenbedingungen in jedem Fall linear unabhängig sind. Unter Beachtung der Tatsache, dass die Maximierung von  $G(\cdot)$  äquivalent zur Minimierung von  $-G(\cdot)$  ist, folgt nach KKT: Es gibt Lagrange-Multiplikatoren  $\lambda_{it} \geq 0$  und  $\mu_t \in \mathbb{R}$ , sodass:

$$\begin{aligned} -\nabla G(x) + \sum_{i=1}^n \sum_{t=1}^T \lambda_{it} \nabla g_{it}(x) + \sum_{t=1}^T \mu_t \nabla h_t(x) &= 0 \\ \lambda_{it} g_{it}(x) &= 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \end{aligned}$$

Die zweite Bedingung ist für  $g_{it}(x) \neq 0$  nur dann erfüllt, wenn  $\lambda_{it} = 0$ . Wir setzen also  $\lambda_{it} = 0$  bei allen inaktiven Ungleichungsnebenbedingungen. Das ergibt:

$$-\nabla G(x) + \sum_{t=1}^T \sum_{i \in I_t(x)} \lambda_{it} \nabla g_{it}(x) + \sum_{t=1}^T \mu_t \nabla h_t(x) = 0$$

Wir haben bereits im Beweis von Lemma 22 festgestellt, dass die Gradienten periodenweise unabhängig sind. Wir betrachten das Ganze also spaltenweise. Für eine beliebige Periode  $t \in \mathcal{T}$  sei OBdA  $I_t(x) = \{1, \dots, m\}$ . Dann:

$$(a) \ i \in I_t(x), x_{it} = 0, \lambda_{it} \geq 0 \quad \left\{ \begin{array}{l} \begin{bmatrix} -\dot{G}_x^{1t} \\ -\dot{G}_x^{2t} \\ \vdots \\ -\dot{G}_x^{mt} \\ -\dot{G}_x^{m+1t} \\ \vdots \\ -\dot{G}_x^{nt} \end{bmatrix} - \begin{bmatrix} \lambda_{1t} \\ \lambda_{2t} \\ \vdots \\ \lambda_{mt} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mu_t \\ \mu_t \\ \vdots \\ \mu_t \\ \mu_t \\ \vdots \\ \mu_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ (b) \ i \notin I_t(x), x_{it} > 0, \lambda_{it} = 0 \end{array} \right.$$

Wir überzeugen uns schnell davon, dass das Subsystem (b) nur dann lösbar ist, wenn:

$$\dot{G}_x^{m+1t} = \dot{G}_x^{m+2t} = \dots = \dot{G}_x^{nt}$$

bzw.

$$\dot{G}_x^{it} = \dot{G}_x^{jt} \quad \forall i \notin I_t(x), j \in I_t(x)$$

womit (62) gezeigt wäre. Um den Beweis abzuschließen, zeigen wir noch (63). Aus dem Subsystem (b) folgern wir, dass

$$\dot{G}_x^{it} = \mu_t \quad \forall i \notin I_t$$

Angenommen, Bedingung (63) würde nicht gelten, also  $\exists j \in I_t(x) : \dot{G}_x^{jt} > \mu_t$ . Dann:

$$-\dot{G}_x^{jt} - \lambda_{jt} + \mu_t = \underbrace{-\dot{G}_x^{jt} + \mu_t}_{<0} - \lambda_{jt} < -\lambda_{jt} \leq 0$$

Also wäre in diesem Fall das Subsystem (a) unlösbar – ein Widerspruch!  $\square$

**Bemerkung 24:** Die Bedingungen (62) und (63) können auch einfacher so formuliert werden: Für jedes  $t \in \mathcal{T}$  gilt:

$$\dot{G}_x^{it} \geq \dot{G}_x^{jt} \quad \forall i \notin I_t(x), j \in \mathcal{N} \quad (64)$$

In Worten formuliert: Betrachten wir eine Spalte  $t \in \mathcal{T}$ , so darf keine Projektklasse einen größeren Anstieg der Zielfunktion bewirken, als jene, in welche wir investieren. Andernfalls wäre ja durch Verschiebungen eine Verbesserung der Zielfunktion möglich.

Der eben aufgeschriebene und bewiesene Satz gilt für den Gradienten  $\dot{G}_x$ . Im folgenden Satz erweitern wir die notwendigen Bedingungen für ein lokales Optimum auf den korrigierten Gradienten  $\ddot{G}_x$ .

**Satz 25:** Sei  $x$  ein lokales Maximum von  $(P)$ . Dann gilt:

$$\ddot{G}_x^{it} = 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T} \quad (65)$$

*Beweis.* Wie schon oft verwendet, sind die Perioden untereinander unabhängig. Wir greifen uns also im Folgenden wiederum eine beliebige Periode  $t \in \mathcal{T}$  heraus und stellen fest, dass das Resultat für alle Perioden gilt.

Nach Satz 23 bzw. Bemerkung 24 gilt für alle  $j \in \mathcal{N}$

$$\dot{G}_x^{jt} \leq \dot{G}_x^{it} \quad \forall i \notin I_t(x) \quad (66)$$

und damit folglich auch für alle  $j \in \mathcal{N}$

$$\dot{G}_x^{jt} \leq \frac{1}{|\mathcal{N}/I_t(x)|} \sum_{i \notin I_t(x)} \dot{G}_x^{it} \quad (67)$$

Das heißt, dass in Algorithmus 3 früher oder später für alle  $j \in \mathcal{N}$  jeweils einer der folgenden beiden Fälle eintritt:

- $\ddot{G}_x^{jt} = 0$  und  $j \notin \dot{\mathcal{N}}_t$ , falls in (67) strikte Ungleichung gilt oder
- $\ddot{G}_x^{jt} = \ddot{G}_x^{it}$  für alle  $i \notin I_t(x)$ , falls (67) mit Gleichheit erfüllt ist.

Zu diesem Zeitpunkt haben wir dann also in Periode  $t$  einerseits nullgesetzte Einträge und andererseits Einträge, die allesamt gleich groß sind. Nach der Mittelwertsbereinigung werden zweitens auch auf Null gesetzt.

Insgesamt gilt also

$$\ddot{G}_x^{it} = 0 \quad \forall i \in \mathcal{N}, t \in \mathcal{T}$$

was zu zeigen war. □

In Satz 20 haben wir gezeigt, dass Algorithmus 3 zulässige Verbesserungsrichtungen liefert, sofern  $\ddot{G}_x \neq 0$  ist. Mit dem eben bewiesenen Satz wissen wir, dass wir im Fall  $\ddot{G}_x = 0$  einen aussichtsreichen Kandidaten für ein lokales Optimum gefunden haben.

## 4.5. Sattelpunkten entfliehen

Erfüllt eine Lösung  $x$  die notwendigen Bedingungen für lokale Optima, so ist dies ein (sehr) starkes Indiz dafür, dass wir auch tatsächlich ein lokales Optimum gefunden haben. Leider ist dies noch keine Garantie dafür; es kann sich bei  $x$  theoretisch auch um einen Sattelpunkt handeln.

Wir überlegen uns jetzt eine Strategie, mit welcher wir Sattelpunkte (fast sicher) ausschließen können<sup>3</sup>. Allerdings hat sich in diversen Versuchen herausgestellt, dass es äußerst unwahrscheinlich sein dürfte, überhaupt einen Sattelpunkt zu erreichen.

Konkret kontrollieren wir separat für jede Periode  $t \in \mathcal{T}$ , wie sich die Zielfunktion verändert, wenn wir ein wenig Arbeitszeit zwischen zwei Projektklassen umschichten, also etwa für zwei Projektklassen  $i$  und  $j$

$$x_{it} \leftarrow x_{it} + \lambda \quad \text{sowie} \quad x_{jt} \leftarrow x_{jt} - \lambda$$

<sup>3</sup>Numerische Ungenauigkeiten können lästig sein!

setzen (so die Lösung zulässig bleibt) und  $G(x)$  auswerten. Dabei ist  $\lambda > 0$  eine kleine Zahl, zB  $\lambda = 10^{-4}$ . Steigt die Zielfunktion, befinden wir uns aktuell in einem Sattelpunkt. Wir springen zur umgeschichteten Lösung und fahren mit der Optimierung fort.

Dies führen wir für alle Projektklassen paarweise aus, das bedeutet, es sind in Summe  $T \cdot n \cdot (n - 1)$  Lösungen zu betrachten. Das können sehr viele sein!

Die gute Nachricht: Sehr viele dieser umgeschichteten Lösungen brauchen wir gar nicht erst zu kontrollieren, da im Vorhinein schon feststeht, dass sie über keinerlei Verbesserungspotenzial verfügen. Für eine Periode  $t \in \mathcal{T}$  können wir alle Projektklassen  $j \in I_t(x)$  von dieser Prozedur ausschließen, für welche gilt:

$$\dot{G}_x^{jt} < \dot{G}_x^{it} \quad \text{für ein } i \notin I_t(x)$$

In der Praxis treten häufig Lösungen auf, die teilweise oder gar gänzlich am Rand liegen, also für einige oder alle Perioden  $t \in \mathcal{T}$

$$|I_t(x)| = n - 1$$

erfüllt ist. Solche Lösungen können wir rasch auf lokale Optimalität hin prüfen.

In [Gut11, Theorem 4, S. 186 f] wurde für den Spezialfall, dass Projektklassen und Kompetenzen zusammenfallen (also jede Projektklasse gleichzeitig ihre eigene Kompetenz ist), unter anderem bewiesen, dass für eine lokal optimale Lösung  $x$

$$1 \leq |\mathcal{N}/I_t(x)| \leq T - t + 1 \quad \forall t \in \mathcal{T}$$

gilt, also in der  $k$ . letzten Periode höchstens in  $k$  Projektklassen investiert wird. Die Vermutung liegt nahe, dass im allgemeinen Fall ein ähnlicher Mechanismus greift.

## 4.6. Linesearch

Mit Hilfe des korrigierten Gradienten können wir für die aktuelle Lösung  $x$  eine zulässige Verbesserungsrichtung ausrechnen. Weiters wissen wir, wann wir abbrechen können.

Ein kleiner aber feiner Puzzleteil fehlt noch. Gegeben einer Lösung  $x$  und der zulässigen Verbesserungsrichtung  $\ddot{G}_x$ : Wie weit sollen wir in diese Richtung gehen? Was ist die optimale Schrittweite?

Diese Frage kann allgemein mit **Linesearch** beantwortet werden. Wir suchen ausgehend von  $x$  entlang der Richtung  $\ddot{G}_x$  nach der optimalen Schrittweite  $\theta^*$  und der entsprechenden Lösung  $x + \theta^* \cdot \ddot{G}_x$ .

Das Startintervall für die Suche ist  $[0, \theta_{max}]$ , wobei die maximale Schrittweite  $\theta_{max}$  wie folgt definiert ist:

$$\theta_{max} = \sup \left\{ \theta \geq 0 : x + \theta \cdot \ddot{G}_x \in \mathcal{S} \right\} \quad (68)$$

Also das größtmögliche  $\theta$ , sodass der zulässige Bereich  $\mathcal{S}$  nicht verlassen wird.

Die Bestimmung von  $\theta_{max}$  ist für  $\ddot{G}_x \neq 0$  einfach:

$$\theta_{max}^- = \min \left\{ -x_{js} / \ddot{G}_x^{js} : j \in \mathcal{N}, s \in \mathcal{T}, \ddot{G}_x^{js} < 0 \right\} \quad (69)$$

$$\theta_{max}^+ = \min \left\{ (\sigma_s - x_{js}) / \ddot{G}_x^{js} : j \in \mathcal{N}, s \in \mathcal{T}, \ddot{G}_x^{js} > 0 \right\} \quad (70)$$

$$\theta_{max} = \min(\theta_{max}^-, \theta_{max}^+) \quad (71)$$

Wenn  $\ddot{G}_x = 0$  gilt, so setzen wir  $\theta_{max} = 0$ . Dieser Umstand tritt in allererster Linie bei Ecklösungen auf.

Es gibt zahlreiche Möglichkeiten der Liniensuche. Wir verwenden in dieser Arbeit die **Methode des Goldenen Schnittes**. Eine übersichtliche Beschreibung dieser Methode findet man zB in [SY06, S. 84 ff].

#### 4.7. Suche nach lokalen Optima

Wir fassen unsere Überlegungen zur Bestimmung eines lokalen Maximums in Algorithmus 4 zusammen.

---

**Algorithmus 4** Berechne kurzfristiges lokales Optimum

---

**Input:**  $x$  – die Startlösung,  $T$  – die Anzahl an Perioden,  $\epsilon > 0$  – die Genauigkeitsschranke,  $\lambda > 0$  – Schrittweite bei Sattelpunktprüfung

**Output:**  $x$  – die lokal optimierte Lösung

```

1: repeat
2:    $\ddot{G}_x \leftarrow$  Berechne den korrigierten Gradienten gemäß Algorithmus 3
3:    $\theta \leftarrow$  Linesearch in Richtung  $\ddot{G}_x$ 
4:    $x \leftarrow x + \theta \cdot \ddot{G}_x$ 
5: until  $\|\ddot{G}_x\|_\infty < \epsilon$  // Jetzt: Prüfe auf Sattelpunkt
6: for  $t \in \mathcal{T}$  do
7:    $\mathcal{I} \leftarrow \mathcal{N} / I_t(x) \cup \left\{ i \in I_t(x) : \left| \dot{G}_x^{it} - \dot{G}_x^{jt} \right| < \epsilon \text{ für ein } j \in \mathcal{N} / I_t(x) \right\}$ 
8:   for  $i \in \mathcal{I}$  do
9:     for  $j \in (\mathcal{N} / I_t(x)) / \{i\}$  do
10:       $x^c \leftarrow x$  // Umschichtung  $j \rightarrow i$ 
11:       $x_{jt}^c \leftarrow \max(x_{jt} - \lambda, 0)$ 
12:       $x_{it}^c \leftarrow x_{it} + (x_{jt} - x_{jt}^c)$ 
13:      if  $G(x^c) > G(x)$  then
14:        Restart mit  $x = x^c$  // Neustart des Algorithmus
15:      end if
16:    end for
17:  end for
18: end for
19: return  $x$ 

```

---

Die Möglichkeiten, die Startlösung zu wählen sind reichlich vorhanden. Wir werden uns diesbezüglich von den nächsten zwei Kapiteln noch inspirieren lassen.

Natürlich ist es immer möglich, eine zufällige Startlösung zu wählen. In diesem Fall lässt sich Algorithmus 4 leicht zu einem **Iterated Local Search** ausdehnen:

- Generiere eine zufällige Startlösung  $x$  gemäß Algorithmus 1 auf Seite 38.
- Führe eine lokale Optimierung gemäß Algorithmus 4 auf der vorherigen Seite durch.

**Beispiel 26 (Lokales Optimierungsbeispiel):** Wir betrachten abschließend ein übersichtliches Beispiel mit  $n = 2$  Projektklassen,  $R = 2$  Kompetenzen und  $T = 2$  Perioden, das aus [Gut11, Beispiel 6, S. 192f] übernommen wurde, um die Theorie dieses Kapitels zu veranschaulichen. Wir verwenden das standardisierte Lernszenario ( $a = b = 1$ ),  $\sigma_t = 1$  für  $t = 1, 2$  sowie

$$d = \begin{bmatrix} 0.85 & 0.15 \\ 0.25 & 0.75 \end{bmatrix} \quad \beta = \begin{pmatrix} 0.2 \\ 0.4 \end{pmatrix} \quad \alpha = \begin{pmatrix} 3 \\ 0.5 \end{pmatrix} \quad w = \begin{pmatrix} 100 \\ 200 \end{pmatrix} \quad z_1 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

Abbildung 9 stellt die Zielfunktion übersichtlich dar und zeigt, dass es 5 besondere Punkte gibt. Außerdem veranschaulichen wir Algorithmus 4.

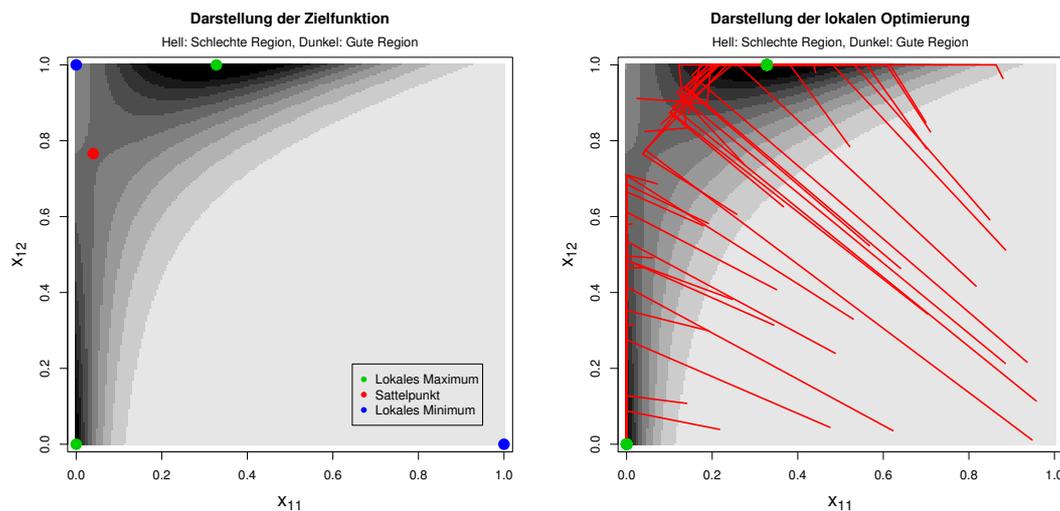


Abbildung 9: Darstellung der Zielfunktion und der lokalen Optimierung im lokalen Optimierungsbeispiel. Die Achsen stehen für die Investitionen in Projektklasse 1 in Periode 1 (x-Achse) und Periode 2 (y-Achse). Klarerweise gilt  $x_{21} = 1 - x_{11}$  und  $x_{22} = 1 - x_{12}$ . **Links:** Darstellung der Zielfunktion, der lokalen Optima und des Sattelpunkts. **Rechts:** Darstellung von 40 Pfaden, welche mit Iterated Local Search erstellt wurden. Sämtliche Pfade konvergieren gegen eines der beiden lokalen Maxima.

In Tabelle 4 listen wir die besonderen 5 Punkte auf und diskutieren diese.

Nr.	$x_{11}$	$x_{12}$	$G(x)$	Charakteristik	$\dot{G}_x^{11}$	$\dot{G}_x^{21}$	$\dot{G}_x^{12}$	$\dot{G}_x^{22}$
1	0.00	0.00	132.59	Lokales Maximum	75.08	85.26	66.98	67.49
2	0.33	1.00	132.63	Lokales Maximum	74.57	74.57	67.35	63.91
3	0.04	0.77	132.20	Sattelpunkt	76.86	76.86	67.08	67.08
4	0.00	1.00	132.08	Lokales Minimum	77.93	74.28	66.98	67.49
5	1.00	0.00	122.19	Lokales Minimum	73.26	83.48	65.47	56.52

Tabelle 4: Darstellung der 5 besonderen Punkte im lokalen Optimierungsbeispiel

Unsere Theorie, welche wir in Satz 23 auf Seite 50 aufgestellt und bewiesen haben, bestätigt sich! Die Gradienten erfüllen für die lokalen Maxima die notwendigen Bedingungen.

In Punkt Nummer 1 sagt uns der Gradient wegen  $\dot{G}_x^{21} > \dot{G}_x^{11}$  und  $\dot{G}_x^{22} > \dot{G}_x^{12}$ , dass wir die Zielfunktion steigern können, wenn wir in beiden Perioden Arbeitszeit von Projektklasse 1 nach Projektklasse 2 verschieben. Das können wir aber nicht tun, da wir sonst den zulässigen Bereich verließen.

In Punkt 2 gilt  $\dot{G}_x^{11} = \dot{G}_x^{21}$ , weshalb unsere gefundene Mischung für Periode 1 die notwendigen Bedingungen erfüllt. Der mittelwertsbereinigte Gradient zeigt in der 2. Periode außerhalb des zulässigen Bereichs, auch hier sind die notwendigen Bedingungen erfüllt.

Die Punkte 4 und 5 sind trivial: Der mittelwertsbereinigte Gradient zeigt ins Innere, daher können beide Punkte nicht lokal maximal sein.

Spannend ist Punkt 3: Er liegt vollständig im Inneren des zulässigen Bereichs und erfüllt  $\dot{G}_x^{11} = \dot{G}_x^{21}$  sowie  $\dot{G}_x^{12} = \dot{G}_x^{22}$ . Trotzdem ist er kein lokales Optimum, da eine Verschiebung um ein kleines  $\lambda > 0$  nach oben

$$x_{12} \leftarrow x_{12} + \lambda \quad \text{sowie} \quad x_{22} \leftarrow x_{22} - \lambda$$

oder unten

$$x_{22} \leftarrow x_{22} + \lambda \quad \text{sowie} \quad x_{12} \leftarrow x_{12} - \lambda$$

zu einer Verbesserung der Zielfunktion führt.

In 100 Iterationen wurde der Sattelpunkt nicht ein Mal erreicht. Die Überprüfung auf Sattelpunkte wäre also nicht notwendig gewesen.

## 5. Langfristige lokale Optimierung

In diesem Kapitel untersuchen wir das Kompetenzentwicklungssystem auf stabile Zustände hin und entwickeln Algorithmen, die uns beim Aufspüren von Lösungen mit langfristigem Optimalitätspotenzial behilflich sind.

Nachdem wir im ersten Unterabschnitt Annahmen formulieren und uns mit der Notation befassen, stellen wir in Abschnitt 5.2 fest, dass Gleichgewichtszustände für das Kompetenzentwicklungssystem existieren, wenn wir in jeder Periode unsere verfügbare Arbeitszeit immer gleich auf die Projektklassen aufteilen und studieren deren Eindeutigkeit.

Im darauf folgenden Abschnitt 5.3 beschreiben wir detailliert, wie wir Gleichgewichtszustände ermitteln können. In Abschnitt 5.4 entwickeln wir einen Algorithmus zur Bestimmung von Portfolios mit langfristigem Optimalitätspotenzial. Abgerundet wird dieses Kapitel mit der Ausdehnung dieses Algorithmus auf ein iteriertes lokales Suchverfahren.

### 5.1. Notation und Annahmen

Es sind in diesem Kapitel einige zusätzliche Annahmen gegenüber dem Basismodell notwendig. Einerseits nehmen wir an, dass die Zeitkonten  $\sigma_t$  und die Erträge  $w_{it}$  projektweise konstant sind, also

$$\sigma_t = \sigma > 0 \quad \forall t \geq 1 \quad (72)$$

$$w_{it} = w_i > 0 \quad \forall t \geq 1, i \in \mathcal{N} \quad (73)$$

Andererseits fordern wir, dass

$$\alpha_r > 0 \quad \forall r \in \mathcal{R} \quad (74)$$

$$\beta_r > 0 \quad \forall r \in \mathcal{R} \quad (75)$$

gilt, es muss also in jeder Kompetenz jederzeit ein Kompetenzabbau stattfinden sowie ein Kompetenzzuwachs möglich sein<sup>4</sup>. Wir bezeichnen mit  $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)^T$  das immerwährende Projektportfolio, das wir oft **Grenzportfolio** nennen:

$$x_t = \bar{x} \quad \forall t \geq 1 \quad (76)$$

Mit  $\mathcal{R}_i$  bezeichnen wir die Menge all jener Kompetenzen, welche für Projektklasse  $i$  erforderlich sind. Weiters sortieren wir jene Kompetenzen aus, die für keine Projektklasse dieses Projektportfolios benötigt werden:

$$\mathcal{R}_i = \{r \in \mathcal{R} : d_{ir} > 0\} \quad \forall i \in \mathcal{N} \quad (77)$$

$$\bar{\mathcal{N}}(\bar{x}) = \{i \in \mathcal{N} : \bar{x}_i > 0\} \quad (78)$$

$$\bar{\mathcal{R}}(\bar{x}) = \{r \in \mathcal{R} : \exists i \in \mathcal{N} : \bar{x}_i > 0 \wedge d_{ir} > 0\} = \bigcup_{i \in \bar{\mathcal{N}}(\bar{x})} \mathcal{R}_i \quad (79)$$

---

<sup>4</sup>Die Ausdehnung auf  $\alpha_r \geq 0$  und  $\beta_r \geq 0 \quad \forall r \in \mathcal{R}$  ist möglich, sie würde aber diese Arbeit an manchen Stellen stark verkomplizieren.

Ein Grenzportfolio  $\bar{x}$  ist **zulässig**, falls

$$\bar{x} \in \bar{\mathcal{S}} = \left\{ \bar{u} \in [0, \sigma]^n : \sum_{i=1}^n \bar{u}_i = \sigma \right\} \quad (80)$$

Wir nehmen an, dass alle Grenzportfolios, welche in diesem Kapitel vorkommen, zulässig sind. Wenn aus dem Zusammenhang klar hervorgeht, welches Grenzportfolio zu Grunde liegt, so schreiben wir  $\bar{\mathcal{N}}$  statt  $\bar{\mathcal{N}}(\bar{x})$  und  $\bar{\mathcal{R}}$  statt  $\bar{\mathcal{R}}(\bar{x})$ .

Wir verwenden in diesem Kapitel eine geringfügig abgeänderte Version der Lernfunktion:

$$\bar{\varphi}(z) := \epsilon_0 + (1 - \epsilon_0) \cdot \varphi(z) = \epsilon_0 + (1 - \epsilon_0) \cdot \frac{1}{1 + a \cdot \exp(-b \cdot z)} \quad (81)$$

Dabei ist  $\epsilon_0 > 0$  eine winzig kleine Zahl (zB  $10^{-10}$ ), welche sicherstellt, dass

$$\gamma_{rt} = \bar{\varphi}(z_{rt}) \geq \epsilon_0 > 0 \quad \forall r \in \mathcal{R}, t \geq 1 \quad (82)$$

gilt. Diese Festlegung hat in der Praxis keinerlei nennenswerten Auswirkungen, wird uns aber in der Theorie vieles erleichtern! Die Gründe sehen wir schon im folgenden Abschnitt.

## 5.2. Gleichgewichtszustände des Kompetenzentwicklungssystems

Jetzt nehmen wir das Kompetenzentwicklungssystem genauer unter die Lupe. Wir betrachten jene Teile des Optimierungsproblems, welche für uns von Interesse sind:

$$z_{rt+1} = z_{rt} - \beta_r + \alpha_r \sum_{i=1}^n d_{irt} \bar{x}_i \quad \forall r \in \mathcal{R}, t \geq 1 \quad (83)$$

$$\gamma_{rt+1} = \bar{\varphi}(z_{rt+1}) = \epsilon_0 + \frac{1 - \epsilon_0}{1 + a \cdot e^{-b \cdot z_{rt+1}}} \quad \forall r \in \mathcal{R}, t \geq 1 \quad (84)$$

Die Startwerte  $z_{r1}$  und  $\gamma_{r1} = \bar{\varphi}(z_{r1})$  sind gegeben. Weiters erinnern wir uns, dass

$$d_{irt} = \frac{d_{ir}}{\gamma_{rt}} \eta_{it} = \frac{d_{ir}}{\gamma_{rt}} \left( \sum_{\rho=1}^R \frac{d_{i\rho}}{\gamma_{\rho t}} \right)^{-1} \in [0, 1] \quad \forall i \in \mathcal{N}, r \in \mathcal{R}, t \geq 1$$

der relative tatsächliche Arbeitsaufwand ist, den wir in Definition 7 auf Seite 27 eingeführt haben und dass folgendes gilt:

$$\sum_{r=1}^R d_{irt} = \sum_{r=1}^R \frac{d_{ir}}{\gamma_{rt}} \left( \sum_{\rho=1}^R \frac{d_{i\rho}}{\gamma_{\rho t}} \right)^{-1} = 1 \quad \forall i \in \mathcal{N}, t \geq 1$$

Gegeben eines Grenzportfolios  $\bar{x}$  interessieren wir uns für einen **Fixpunkt** bzw. **Gleichgewichtszustand** des Gleichungssystems (83) und (84), also für Effizienzwerte, die sich

von einer Periode zur nächsten nicht ändern, wenn wir das Grenzportfolio  $\bar{x}$  anwenden. Wir bezeichnen mit  $\bar{\gamma} = (\bar{\gamma}_1, \bar{\gamma}_2, \dots, \bar{\gamma}_R)^T$  diesen Fixpunkt.

Wir formulieren Gleichung (83) wie folgt um:

$$\Delta z_{rt} = z_{rt+1} - z_{rt} = -\beta_r + \alpha_r \sum_{i=1}^n \frac{d_{ir}}{\gamma_{rt}} \eta_{it} \bar{x}_i \quad \forall r \in \mathcal{R}, t \geq 1 \quad (85)$$

Und führen folgende Konstrukte ein:

$$\bar{\eta}_i = \left( \sum_{r \in \mathcal{R}_i} \frac{d_{ir}}{\bar{\gamma}_r} \right)^{-1} \quad \bar{d}_{ir} = \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \quad \Delta \bar{z}_r = -\beta_r + \alpha_r \sum_{i \in \mathcal{N}} \bar{d}_{ir} \bar{x}_i \quad (86)$$

Die Vektoren  $\bar{\gamma}$ ,  $\bar{\eta} = (\bar{\eta}_1, \bar{\eta}_2, \dots, \bar{\eta}_n)^T$  und  $\Delta \bar{z} = (\Delta \bar{z}_1, \Delta \bar{z}_2, \dots, \Delta \bar{z}_R)^T$  sowie die Werte für alle  $\bar{d}_{ir}$  hängen vom zugrunde liegenden Grenzportfolio  $\bar{x}$  ab. Wir verzichten auf die Möglichkeit, dies explizit darzustellen.

Wir weisen darauf hin, dass  $\Delta \bar{z}_r$  auch eine Funktion in  $\bar{\gamma}$  ist. Mit  $\Delta \bar{z}_r(\bar{\gamma})$  notieren wir im Bedarfsfall die Auswertung dieser Funktion an der Stelle  $\bar{\gamma}$ .

### 5.2.1. Stabilitätsbedingungen für Kompetenzen

Kommende Definition zeigt uns, wie ein Fixpunkt von (83) und (84) unter Zuhilfenahme von Gleichung (85) aussehen muss.

**Definition 27:** Der Vektor  $\bar{\gamma} \in [\epsilon_0, 1]^R$  ist genau dann ein Fixpunkt von (83)-(84), wenn auf jede Kompetenz  $r \in \mathcal{R}$  genau eine der folgenden drei Möglichkeiten zutrifft:

$$\text{Stabilität A: } \bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0 \quad (87)$$

$$\text{Stabilität B: } \bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0 \quad (88)$$

$$\text{Stabilität C: } \bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0 \quad (89)$$

Die maximale Effizienz  $\bar{\gamma}_r = 1$  kann nur erreicht werden, wenn der entsprechende Kompetenzscore  $z_{rt}$  gegen unendlich geht, was gleichbedeutend mit  $\Delta \bar{z}_r > 0$  ist. Gleiches gilt für Stabilität C mit umgekehrten Vorzeichen. Ist  $\bar{\gamma}_r \in (\epsilon_0, 1)$ , so darf sich der Kompetenzscore wegen  $\gamma_{rt} = \bar{\varphi}(z_{rt})$  nicht ändern. Es muss also  $\Delta \bar{z}_r = 0$  gelten.

Die beiden Sonderfälle, wo  $\bar{\gamma}_r \in \{\epsilon_0, 1\}$  und gleichzeitig  $\Delta \bar{z}_r = 0$  gilt, sind praxisfremd, müssen aber der Sauberkeit halber mitberücksichtigt werden.

Wir klären jetzt auf, warum wir die Lernfunktion abgeändert haben. Angenommen wir würden die gewöhnliche Lernfunktion  $\varphi(z)$  heranziehen. Dann könnte im Fixpunkt  $\bar{\gamma}_r = 0$  für einige  $r \in \mathcal{R}$  gelten. In diesem Fall wären allerdings die Ausdrücke

$$\bar{d}_{ir} = \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \quad \text{sowie} \quad \Delta \bar{z}_r = -\beta_r + \alpha_r \sum_{i \in \mathcal{N}} \bar{d}_{ir} \bar{x}_i$$

nicht definiert, da die Division  $0/0$  auftritt. Dank unserer Festlegung umschiffen wir diese leidige Debatte, ohne einen nennenswerten, praxisrelevanten Fehler zu produzieren!

## 5.2.2. Zusammenhängende Grenzportfolios

Zunächst schreiben wir zwei wesentliche Definitionen auf.

**Definition 28:** Die Menge  $\bar{\mathcal{N}}(\bar{x})$  heißt **zusammenhängend**, falls es nicht möglich ist, sie derart in zwei disjunkte nichtleere Mengen  $\bar{\mathcal{N}}^{(1)}$  und  $\bar{\mathcal{N}}^{(2)} = \bar{\mathcal{N}}/\bar{\mathcal{N}}^{(1)}$  aufzuteilen, dass

$$\bigcup_{i \in \bar{\mathcal{N}}^{(1)}} \mathcal{R}_i \cap \bigcup_{i \in \bar{\mathcal{N}}^{(2)}} \mathcal{R}_i = \emptyset \quad (90)$$

Mit anderen Worten: Gegeben  $\bar{\mathcal{N}}^{(1)}$  und  $\bar{\mathcal{N}}^{(2)}$  darf die Bearbeitung der Projektklassen nicht zu zwei unabhängigen Inselbetrieben führen, was den Einsatz der in diesen Mengen erforderlichen Kompetenzen angeht.

**Definition 29:** Eine Menge  $\bar{\mathcal{N}}_{IB} \subseteq \bar{\mathcal{N}}$  nennen wir (**minimalen**) **Inselbetrieb**, wenn sie zusammenhängend gemäß Definition 28 ist.

Die Liste aller minimalen Inselbetriebe kann einfach mit Hilfe von Algorithmus 5 bestimmt werden. Besteht diese Liste aus nur einem Element – nämlich  $\bar{\mathcal{N}}$  – so ist die Menge  $\bar{\mathcal{N}}$  zusammenhängend.

**Beispiel 30:** Ein einfaches Beispiel mit  $R = 3$  Kompetenzen,  $n = 4$  Projektklassen und  $\sigma = 1$ .

$$d = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1 & 0 \end{bmatrix} \quad \bar{x}^1 = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} \quad \bar{x}^2 = \begin{pmatrix} 1/3 \\ 1/3 \\ 0 \\ 1/3 \end{pmatrix} \quad \bar{x}^3 = \begin{pmatrix} 1/3 \\ 0 \\ 1/3 \\ 1/3 \end{pmatrix}$$

Aus der Matrix  $d$  lassen sich schnell die Mengen  $\mathcal{R}_i$  bestimmen:

$$\mathcal{R}_1 = \{1\} \quad \mathcal{R}_2 = \{1, 2\} \quad \mathcal{R}_3 = \{2, 3\} \quad \mathcal{R}_4 = \{2\}$$

Und aus den Vektoren  $\bar{x}^1$ ,  $\bar{x}^2$  und  $\bar{x}^3$  rasch die Mengen

$$\bar{\mathcal{N}}^1 = \bar{\mathcal{N}}(\bar{x}^1) = \{1, 2, 3, 4\} \quad \bar{\mathcal{N}}^2 = \bar{\mathcal{N}}(\bar{x}^2) = \{1, 2, 4\} \quad \bar{\mathcal{N}}^3 = \bar{\mathcal{N}}(\bar{x}^3) = \{1, 3, 4\}$$

Die Menge  $\bar{\mathcal{N}}^1$  ist zusammenhängend. Am einfachsten wählen wir  $i = 2$  in Zeile 3 von Algorithmus 5. Dann gilt:

$$\mathcal{R}_1 \cap \bar{\mathcal{R}}_{IB} = \{1\} \quad \mathcal{R}_3 \cap \bar{\mathcal{R}}_{IB} = \{2\} \quad \mathcal{R}_4 \cap \bar{\mathcal{R}}_{IB} = \{2\}$$

Folglich benötigt jede Projektklasse mind. eine Kompetenz, welche auch für Projektklasse 2 erforderlich ist und Algorithmus 5 gibt uns  $\mathcal{M} = \{\{1, 2, 3, 4\}\} = \{\bar{\mathcal{N}}\}$  zurück.

---

**Algorithmus 5** Bestimme alle minimalen Inselbetriebe

---

**Input:**  $\bar{\mathcal{N}}(\bar{x})$  – Menge aller zu bearbeitenden Projektklassen im zugrunde liegenden zulässigen Grenzportfolio  $\bar{x}$

**Output:**  $\mathcal{M}$  – Liste aller minimalen Inselbetriebe

```
1:  $\bar{\mathcal{N}}_E \leftarrow \bar{\mathcal{N}}$  // Noch einzufügende Projektklassen
2: while  $\bar{\mathcal{N}}_E \neq \emptyset$  do
3:   Wähle  $i \in \bar{\mathcal{N}}_E$  beliebig
4:    $\bar{\mathcal{N}}_{IB} \leftarrow \{i\}$  // Alle Projektklassen des aktuellen Inselbetriebs
5:    $\bar{\mathcal{R}}_{IB} \leftarrow \mathcal{R}_i$  // Alle für den Inselbetrieb erforderlichen Kompetenzen
6:   loop
7:      $\bar{\mathcal{N}}_{IB}^+ \leftarrow \{j \in \bar{\mathcal{N}}_E / \bar{\mathcal{N}}_{IB} : \mathcal{R}_j \cap \bar{\mathcal{R}}_{IB} \neq \emptyset\}$ 
8:     if  $\bar{\mathcal{N}}_{IB}^+ = \emptyset$  then
9:       Füge die Menge  $\bar{\mathcal{N}}_{IB}$  der Liste  $\mathcal{M}$  hinzu
10:       $\bar{\mathcal{N}}_E \leftarrow \bar{\mathcal{N}}_E / \bar{\mathcal{N}}_{IB}$ 
11:      Break
12:     end if
13:      $\bar{\mathcal{N}}_{IB} \leftarrow \bar{\mathcal{N}}_{IB} \cup \bar{\mathcal{N}}_{IB}^+$ 
14:      $\bar{\mathcal{R}}_{IB} \leftarrow \bar{\mathcal{R}}_{IB} \cup \mathcal{R}_j \quad \forall j \in \bar{\mathcal{N}}_{IB}^+$ 
15:   end loop
16: end while
17: return  $\mathcal{M}$ 
```

---

Die Menge  $\bar{\mathcal{N}}^2$  ist ebenfalls zusammenhängend:  $\mathcal{M} = \{\{1, 2, 4\}\} = \{\bar{\mathcal{N}}\}$ . In diesem Fall ist  $\bar{\mathcal{R}} = \{1, 2\} \subset \{1, 2, 3\} = \mathcal{R}$ . Das heißt, Kompetenz 3 wird nicht benötigt.

Die Menge  $\bar{\mathcal{N}}^3$  ist nicht zusammenhängend. Algorithmus 5 gibt uns

$$\mathcal{M} = \{\bar{\mathcal{N}}_{IB}^{(1)}, \bar{\mathcal{N}}_{IB}^{(2)}\} = \{\{1\}, \{3, 4\}\}$$

zurück und es gilt:

$$\begin{aligned}\bar{\mathcal{N}}_{IB}^{(1)} \cup \bar{\mathcal{N}}_{IB}^{(2)} &= \{1, 3, 4\} = \bar{\mathcal{N}} \\ \bar{\mathcal{N}}_{IB}^{(1)} \cap \bar{\mathcal{N}}_{IB}^{(2)} &= \emptyset\end{aligned}$$

Da der Schnitt der beiden Mengen

$$\bigcup_{i \in \bar{\mathcal{N}}_{IB}^{(1)}} \mathcal{R}_i = \{1\} \quad \text{und} \quad \bigcup_{i \in \bar{\mathcal{N}}_{IB}^{(2)}} \mathcal{R}_i = \{2, 3\}$$

leer ist, haben wir nachgewiesen, dass  $\bar{\mathcal{N}}^3$  nicht zusammenhängend ist. In Abbildung 10 auf der nächsten Seite veranschaulichen wir diese drei Situationen.

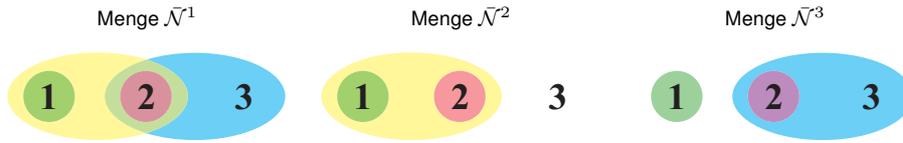


Abbildung 10: Darstellung der Inselbetriebe in Beispiel 30. Die Zahlen stehen für die Kompetenzen und die Ellipsen kennzeichnen die Projektklassen, welche bearbeitet werden. Grün steht für Projektklasse 1, Gelb für 2, Blau für 3 und Rot für Klasse 4. **Links & Mitte:** Astrein zusammenhängend. **Rechts:** Man erkennt sehr schön die beiden unabhängigen Inselbetriebe.

### 5.2.3. Skalierungsinvarianz und Monotonieeigenschaft

Wir betrachten für ein  $r \in \mathcal{R}$  die Gleichung

$$\begin{aligned} \Delta \bar{z}_r &= -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \bar{d}_{ir} \bar{x}_i = -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \bar{x}_i \\ &= -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \frac{d_{ir}}{\bar{\gamma}_r} \left( \sum_{\rho \in \mathcal{R}_i} \frac{d_{i\rho}}{\bar{\gamma}_\rho} \right)^{-1} \bar{x}_i = -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} d_{ir} \left( \sum_{\rho \in \mathcal{R}_i} d_{i\rho} \frac{\bar{\gamma}_r}{\bar{\gamma}_\rho} \right)^{-1} \bar{x}_i \end{aligned}$$

Frage: Wie verändern sich  $\bar{d}_{ir}$  bzw.  $\Delta \bar{z}_r$ , wenn wir  $\bar{\gamma}$  um den Faktor  $q > 0$  skalieren, also  $\bar{\gamma}_r \leftarrow \bar{\gamma}_r \cdot q$  für alle  $r \in \mathcal{R}$  setzen? Offensichtliche Antwort: Gar nicht, da sich der Skalierungsfaktor wegekürzt.

Das heißt,  $\Delta \bar{z}$  ist invariant gegenüber Skalierungen von  $\bar{\gamma}$ . Wir nennen diese Eigenschaft auch kurz **Skalierungsinvarianz**.

Wir wenden uns der Dynamik des Kompetenzentwicklungssystems zu. Die Ableitungen

$$\frac{\partial \bar{\eta}_i}{\partial \bar{\gamma}_r} = \frac{\partial}{\partial \bar{\gamma}_r} \left( \sum_{\rho \in \mathcal{R}_i} \frac{d_{i\rho}}{\bar{\gamma}_\rho} \right)^{-1} = - \left( \sum_{\rho \in \mathcal{R}_i} \frac{d_{i\rho}}{\bar{\gamma}_\rho} \right)^{-2} \cdot \frac{\partial d_{ir}/\bar{\gamma}_r}{\partial \bar{\gamma}_r} = \bar{\eta}_i^2 d_{ir} \frac{1}{\bar{\gamma}_r^2} = d_{ir} \frac{\bar{\eta}_i^2}{\bar{\gamma}_r^2}$$

und die Tatsache, dass unter unserer Redefinition der Lernfunktion von oben stets

$$\bar{d}_{ir} = \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \in [0, 1]$$

wohldefiniert ist, helfen uns dabei, die partiellen Ableitungen von  $\Delta \bar{z}_r$  zu berechnen.

$$\begin{aligned} \frac{\partial \Delta \bar{z}_r}{\partial \bar{\gamma}_r} &= \alpha_r \sum_{i \in \mathcal{N}} d_{ir} \bar{\eta}_i \bar{x}_i \left( -\frac{1}{\bar{\gamma}_r^2} \right) + \frac{d_{ir}}{\bar{\gamma}_r} \bar{x}_i \left( d_{ir} \frac{\bar{\eta}_i^2}{\bar{\gamma}_r^2} \right) = \frac{\alpha_r}{\bar{\gamma}_r} \sum_{i \in \mathcal{N}} -\bar{x}_i \bar{d}_{ir} + \bar{x}_i \bar{d}_{ir}^2 \\ &= \frac{\alpha_r}{\bar{\gamma}_r} \sum_{i \in \mathcal{N}} \bar{x}_i (\bar{d}_{ir}^2 - \bar{d}_{ir}) \leq 0 \quad \forall r \in \mathcal{R} \end{aligned} \quad (91)$$

$$\frac{\partial \Delta \bar{z}_r}{\partial \bar{\gamma}_\rho} = \alpha_r \sum_{i \in \mathcal{N}} \frac{d_{ir}}{\bar{\gamma}_r} \bar{x}_i d_{i\rho} \frac{\bar{\eta}_i^2}{\bar{\gamma}_\rho^2} = \frac{\alpha_r}{\bar{\gamma}_\rho} \sum_{i \in \mathcal{N}} \bar{x}_i \bar{d}_{ir} \bar{d}_{i\rho} \geq 0 \quad \forall r, \rho \in \mathcal{R}, \rho \neq r \quad (92)$$

Wir sehen also, dass  $\Delta \bar{z}_r$  monoton fallend in  $\bar{\gamma}_r$  und monoton wachsend in  $\bar{\gamma}_\rho$  für  $\rho \neq r$  ist. Diese **Monotonieeigenschaft** ist für den Nachweis der Existenz sowie die Analyse der Eindeutigkeit von Fixpunkten von zentraler Bedeutung! Wenn wir also  $\bar{\gamma}_r$  erhöhen, so sinkt  $\Delta \bar{z}_r$  (oder bleibt gleich) und alle  $\Delta \bar{z}_\rho$  mit  $\rho \neq r$  steigen (oder bleiben gleich).

**Einfache Folgerung:** Die Monotonieeigenschaft gilt auch in folgendem simultanen Sinn: Sei  $\dot{\mathcal{R}} \subset \mathcal{R}$ . Wenn wir alle Effizienzwerte  $\bar{\gamma}_r$  der Menge  $\dot{\mathcal{R}}$  erhöhen und jene der Menge  $\mathcal{R}/\dot{\mathcal{R}}$  in Ruhe lassen, so erhöhen sich alle  $\Delta \bar{z}_\rho$  der Menge  $\mathcal{R}/\dot{\mathcal{R}}$  (oder bleiben gleich). Gleiches gilt in umgekehrter Richtung.

**Bemerkung 31:** Die Skalierungsinvarianz und Monotonieeigenschaft gelten für beliebige  $\bar{\gamma} > 0$ . Also auch dann, wenn  $0 < \bar{\gamma}_r < \epsilon_0$  oder  $\bar{\gamma}_r > 1$  für einige  $r \in \mathcal{R}$  wäre. Diese Beobachtung kommt im übernächsten Unterabschnitt oft zur Geltung.

#### 5.2.4. Existenz von Gleichgewichtszuständen

Bevor wir in Satz 33 zeigen, dass ein Gleichgewicht tatsächlich existiert, schreiben wir ein wohlbekanntes Lemma auf, das aus [Bom98] übernommen wurde.

**Lemma 32:** Sei  $(a_n)_{n \in \mathbb{N}}$  eine Folge. Wenn  $(a_n)_{n \in \mathbb{N}}$  beschränkt und monoton ist, so konvergiert  $(a_n)_{n \in \mathbb{N}}$ .

**Satz 33:** Sei  $\bar{x}$  ein zulässiges Grenzportfolio. Dann gilt: Das Gleichungssystem (83) und (84) besitzt einen Fixpunkt  $\bar{\gamma}$ .

*Beweis.* Wir konstruieren eine Folge  $\bar{\gamma}^{(k)}$  beginnend mit dem Startvektor  $\bar{\gamma}^{(1)} \in [\epsilon_0, 1]^R$ . Ausgehend vom  $k$ . Folgenglied bestimmen wir nun mit Hilfe der Prozedur  $\mathcal{P}$ , welche wir gleich im Anschluss aufschreiben, das  $k+1$ . Folgenglied, also

$$\bar{\gamma}^{(k+1)} = \mathcal{P}(\bar{\gamma}^{(k)}) \quad (93)$$

Prozedur  $\mathcal{P}$  in Iteration  $k$ :

- Setze  $\bar{\gamma}^{(k+1)} = \bar{\gamma}^{(k)}$  und wähle  $r = r(k) = (k-1) \pmod{R} + 1$ .
- Falls  $\Delta \bar{z}_r(\bar{\gamma}^{(k+1)}) < 0$ : Senke  $\bar{\gamma}_r^{(k+1)}$  so lange, bis  $\Delta \bar{z}_r(\bar{\gamma}^{(k+1)}) = 0$  erfüllt ist. Ist im Intervall  $[\epsilon_0, \bar{\gamma}_r^{(k)}]$  kein solcher Wert auffindbar, so setze  $\bar{\gamma}_r^{(k+1)} = \epsilon_0$ .

Wir gehen also in der Prozedur  $\mathcal{P}$  die Kompetenzen zyklisch durch und senken in jeder Iteration gegebenenfalls genau einen Effizienzwert. Nach jeweils  $R$  Durchgängen beginnen wir wieder bei der ersten Kompetenz. Wegen der Monotonieeigenschaft ist die Bestimmung von  $\bar{\gamma}^{(k+1)}$  sehr einfach möglich.

Offensichtlich erzeugt  $\mathcal{P}$  für jeden Startvektor  $\bar{\gamma}^{(1)}$  eine monoton fallende Folge  $\bar{\gamma}^{(k)}$ , also

$$\bar{\gamma}_r^{(k+1)} \leq \bar{\gamma}_r^{(k)} \quad \forall r \in \mathcal{R}, k \geq 1$$

Weiters ist diese Folge nach unten durch  $\epsilon_0$  beschränkt. Nach Lemma 32 existiert also für jeden Startvektor ein Grenzwert

$$\bar{\gamma}^* = \lim_{k \rightarrow \infty} \bar{\gamma}^{(k)} \quad (94)$$

Wir verwenden ab jetzt den Startvektor  $\bar{\gamma}^{(1)} = (1, 1, \dots, 1)^T$  und zeigen, dass der Grenzwert  $\bar{\gamma}^*$  für diesen Startvektor ein Fixpunkt gemäß Definition 27 sein muss.

Dazu überzeugen wir uns bei Betrachtung der Prozedur  $\mathcal{P}$  davon, dass  $\Delta \bar{z}_r(\bar{\gamma}) > 0$  nur dann möglich ist, wenn  $\bar{\gamma}_r = 1$  ist. Denn sobald  $\Delta \bar{z}_r < 0$  für ein  $r \in \mathcal{R}$  erstmalig eintritt, wird  $\bar{\gamma}_r$  in weiterer Folge höchstens soweit abgesenkt, bis  $\Delta \bar{z}_r = 0$  gilt. Und eine Senkung von  $\bar{\gamma}_\rho$  für  $\rho \neq r$  bewirkt eine Senkung von  $\Delta \bar{z}_r$ . Somit gilt für alle  $k \in \mathbb{N}$ :

$$\Delta \bar{z}_r(\bar{\gamma}^{(k)}) \leq 0 \quad \forall r \in \mathcal{R} : \bar{\gamma}_r^{(k)} < 1 \quad (95)$$

**Teil 1:** Angenommen, es gibt  $r \in \mathcal{R}$ , sodass  $\bar{\gamma}_r^* < 1 \wedge \Delta \bar{z}_r(\bar{\gamma}^*) > 0$

Wir beobachten, dass  $\Delta \bar{z}_r$  eine stetige Funktion in  $\bar{\gamma}_r$  ist. Aus der Monotonieeigenschaft und wegen der Stetigkeit von  $\Delta \bar{z}_r$  folgt nun, dass es ein  $\epsilon > 0$  und ein Intervall  $[\bar{\gamma}_r^*, \bar{\gamma}_r^* + \epsilon]$  gibt, sodass  $\Delta \bar{z}_r((\bar{\gamma}_1^*, \dots, \bar{\gamma}_{r-1}^*, \bar{\gamma}_r, \bar{\gamma}_{r+1}^*, \dots, \bar{\gamma}_R^*)^T) > 0$  gilt für alle  $\bar{\gamma}_r \in [\bar{\gamma}_r^*, \bar{\gamma}_r^* + \epsilon]$ .

Da  $\bar{\gamma}^*$  Grenzwert der von  $\mathcal{P}$  für den Startwert  $\bar{\gamma}^{(1)} = (1, 1, \dots, 1)^T$  erzeugten Folge  $\bar{\gamma}^{(k)}$  ist, kommt  $\bar{\gamma}^{(k)}$  diesem Grenzwert beliebig nahe. Insbesondere gibt es  $K \in \mathbb{N}$ , sodass

$$\bar{\gamma}_r^{(k)} < 1 \wedge \left\| \bar{\gamma}^{(k)} - \bar{\gamma}^* \right\|_\infty < \epsilon \quad \forall k \geq K$$

Durch gerade getätigte Überlegungen bleibt  $\Delta \bar{z}_r(\bar{\gamma}^*) > 0$ , wenn wir  $\bar{\gamma}_r^*$  auf  $\bar{\gamma}_r^{(K)}$  erhöhen. Aus der Monotonieeigenschaft folgt nun, dass sich das Vorzeichen von  $\Delta \bar{z}_r$  erst recht nicht ändert, wenn wir  $\bar{\gamma}_\rho^*$  auf  $\bar{\gamma}_\rho^{(K)}$  anheben für  $\rho \neq r$ . Unterm Strich gilt also:

$$\Delta \bar{z}_r(\bar{\gamma}^{(K)}) > 0$$

Nach  $K$  Iterationen liefert uns  $\mathcal{P}$  also einen Zustand, in dem gleichzeitig  $\bar{\gamma}_r^{(K)} < 1$  und  $\Delta \bar{z}_r(\bar{\gamma}^{(K)}) > 0$  gilt. Das kann aber für den Startwert  $\bar{\gamma}^{(1)} = (1, 1, \dots, 1)^T$  nicht passieren! Denn es gilt ja  $\bar{\gamma}_r^{(k)} < 1$  für alle  $k \geq K$  und gemäß (95) gilt für alle  $k \geq K$  auch

$$\Delta \bar{z}_r(\bar{\gamma}^{(k)}) \leq 0$$

Ein Widerspruch zu unserer Annahme.

**Teil 2:** Angenommen, es gibt  $r \in \mathcal{R}$ , sodass  $\bar{\gamma}_r^* > \epsilon_0 \wedge \Delta \bar{z}_r(\bar{\gamma}^*) < 0$

Da  $\Delta \bar{z}_r$  stetig in allen Effizienzwerten  $\bar{\gamma}_\rho$  ist, folgt

$$0 > \Delta \bar{z}_r(\bar{\gamma}^*) = \lim_{k \rightarrow \infty} \Delta \bar{z}_r(\bar{\gamma}^{(k)})$$

Wiederum können wir diesem Grenzwert beliebig nahe kommen. Insbesondere gibt es  $K \in \mathbb{N}$ , sodass

$$\Delta \bar{z}_r(\bar{\gamma}^{(k)}) < 0 \wedge \bar{\gamma}_r^{(k)} > \epsilon_0 \quad \forall k \geq K$$

gilt. Das widerspricht jedoch der Prozedur  $\mathcal{P}$ . Denn diese Prozedur würde nach längstens  $K + R$  Iterationen durch Senken von  $\bar{\gamma}_r$  entweder den Wert  $\Delta \bar{z}_r$  auf 0 anheben, oder  $\bar{\gamma}_r = \epsilon_0$  setzen.

Folglich gilt für den Startvektor  $\bar{\gamma}^{(1)} = (1, 1, \dots, 1)^T$ :

$$\begin{aligned} \Delta \bar{z}_r(\bar{\gamma}^*) &\leq 0 \quad \forall r \in \mathcal{R} : \bar{\gamma}_r^* < 1 \\ \Delta \bar{z}_r(\bar{\gamma}^*) &\geq 0 \quad \forall r \in \mathcal{R} : \bar{\gamma}_r^* > \epsilon_0 \end{aligned}$$

oder umformuliert:

$$\begin{aligned} \forall r \in \mathcal{R} : \bar{\gamma}_r^* = 1 &\text{ gilt: } \Delta \bar{z}_r(\bar{\gamma}^*) \geq 0 \text{ (Stabilität A oder B)} \\ \forall r \in \mathcal{R} : \bar{\gamma}_r^* \in (\epsilon_0, 1) &\text{ gilt: } \Delta \bar{z}_r(\bar{\gamma}^*) = 0 \text{ (Stabilität B)} \\ \forall r \in \mathcal{R} : \bar{\gamma}_r^* = \epsilon_0 &\text{ gilt: } \Delta \bar{z}_r(\bar{\gamma}^*) \leq 0 \text{ (Stabilität B oder C)} \end{aligned}$$

Also erfüllt für den Vektor  $\bar{\gamma}^*$  jede Kompetenz entweder Stabilität A, B oder C gemäß Definition 27, was zu zeigen war.  $\square$

Das Konzept, das wir im Beweis verwendet haben, wird Ihnen, werte Leserin und werter Leser, weiter unten bei der Berechnung des Gleichgewichtszustands bekannt vorkommen!

### 5.2.5. Eindeutigkeit von Gleichgewichtszuständen

Wir wissen also, dass Gleichgewichtszustände existieren. Angenommen, wir haben einen solchen gefunden: Wie sieht es mit der Eindeutigkeit aus? Dieser Frage wenden wir uns in diesem Unterabschnitt zu.

Bevor wir mit einer Satzatterie Klarheit schaffen, bemerken wir zunächst, dass wir für ein gegebenes Grenzportfolio  $\bar{x}$  oft nicht alle Kompetenzen benötigen. Es gilt

$$\bar{\mathcal{R}} \subseteq \mathcal{R} \tag{96}$$

Falls eine echte Teilmengenbeziehung vorliegt, können wir wegen unserer Kapitelannahme

$$\beta_r > 0 \quad \forall r \in \mathcal{R}$$

für jede unbenötigte Kompetenz  $r \in \mathcal{R}/\bar{\mathcal{R}}$  sofort den eindeutigen Wert für  $\bar{\gamma}_r$  ermitteln:

$$\bar{\gamma}_r = \epsilon_0 \quad \forall r \in \mathcal{R}/\bar{\mathcal{R}} \quad (97)$$

In allen Sätzen dieses Unterabschnitts bezieht sich  $\Delta\bar{z}$  auf den aktuellen Zustand von  $\bar{\gamma}$ .

**Satz 34 (Eindeutigkeitssatz A):** Sei  $\bar{\gamma}$  ein Fixpunkt, in dem für jede benötigte Kompetenz Stabilität A ( $\bar{\gamma}_r = 1 \wedge \Delta\bar{z}_r > 0 \quad \forall r \in \mathcal{R}$ ) gilt. Dann ist dieser Fixpunkt eindeutig.

*Beweis.* Obiges Argument besagt, dass wir für alle nicht benötigten Kompetenzen aus der Menge  $\mathcal{R}/\bar{\mathcal{R}}$  sofort den eindeutigen Wert für  $\bar{\gamma}_r$  bestimmen können. Dh wir können, um uns mühsame Indexschlachten zu ersparen, OBdA annehmen, dass sämtliche Kompetenzen gebraucht werden, dass also  $\bar{\mathcal{R}} = \mathcal{R}$  gilt.

Sei  $\bar{\gamma} \neq \bar{\gamma}$  ein beliebiger Vektor in  $[\epsilon_0, 1]^R$  und sei  $r_{min} = \arg \min_{r \in \bar{\mathcal{R}}} \bar{\gamma}_r$ .

Wir bemerken, dass wegen  $\bar{\gamma} \neq \bar{\gamma}$  jedenfalls  $\bar{\gamma}_{r_{min}} < 1$  gelten muss. Gegeben, dass  $\bar{\gamma}$  ein Gleichgewicht ist, zeigen wir in zwei Schritten, dass  $\bar{\gamma}$  kein Gleichgewicht sein kann.

Schritt 1: Wir **skalieren** alle  $\bar{\gamma}_r$  um  $\bar{\gamma}_{r_{min}}$ , dh, wir setzen alle  $\bar{\gamma}_r$  auf  $\bar{\gamma}_r \cdot \bar{\gamma}_{r_{min}} = \bar{\gamma}_{r_{min}}$ . Wegen der Skalierungsinvarianz bleibt  $\Delta\bar{z}$  unverändert. Es gilt also nach wie vor  $\Delta\bar{z}_r > 0$  für alle  $r \in \bar{\mathcal{R}}$ . Insbesondere herrscht  $\bar{\gamma}_r \leq \bar{\gamma}_r$  für alle  $r \in \bar{\mathcal{R}}$  vor.

Schritt 2: Wir **erhöhen**  $\bar{\gamma}_r$  auf  $\bar{\gamma}_r$  für alle  $r \neq r_{min}$ . Da  $\bar{\gamma}_{r_{min}} = \bar{\gamma}_{r_{min}}$  bereits nach Schritt 1 vorherrscht, ist in dieser Kompetenz keine Anpassung mehr notwendig. Wegen der Monotonieeigenschaft gilt jetzt immer noch  $\Delta\bar{z}_{r_{min}} > 0$ . Da gleichzeitig  $\bar{\gamma}_{r_{min}} < 1$  ist, kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein.

In Abbildung 11 auf der nächsten Seite wird die Situation veranschaulicht. □

**Satz 35 (Eindeutigkeitssatz C):** Sei  $\bar{\gamma}$  ein Fixpunkt, in dem für jede benötigte Kompetenz Stabilität C ( $\bar{\gamma}_r = \epsilon_0 \wedge \Delta\bar{z}_r < 0 \quad \forall r \in \bar{\mathcal{R}}$ ) gilt. Dann ist dieser Fixpunkt eindeutig.

*Beweis.* Völlig analog zum Beweis des vorangehenden Satzes. Sei wieder OBdA  $\bar{\mathcal{R}} = \mathcal{R}$ . Sei  $\bar{\gamma} \neq \bar{\gamma}$  ein beliebiger Vektor in  $[\epsilon_0, 1]^R$  und sei  $r_{max} = \arg \max_{r \in \bar{\mathcal{R}}} \bar{\gamma}_r$ .

Wir bemerken, dass wegen  $\bar{\gamma} \neq \bar{\gamma}$  jedenfalls  $\bar{\gamma}_{r_{max}} > \epsilon_0$  gelten muss. Gegeben, dass  $\bar{\gamma}$  ein Gleichgewicht ist, zeigen wir in zwei Schritten, dass  $\bar{\gamma}$  kein Gleichgewicht sein kann.

Schritt 1: Wir **skalieren** alle  $\bar{\gamma}_r$  um  $\bar{\gamma}_{r_{max}}/\epsilon_0$ . Wegen der Skalierungsinvarianz bleibt  $\Delta\bar{z}$  unverändert. Es gilt also nach wie vor  $\Delta\bar{z}_r < 0$  für alle  $r \in \bar{\mathcal{R}}$ . Insbesondere herrscht  $\bar{\gamma}_r \geq \bar{\gamma}_r$  für alle  $r \in \bar{\mathcal{R}}$  vor.

Schritt 2: Wir **senken**  $\bar{\gamma}_r$  auf  $\bar{\gamma}_r$  für alle  $r \neq r_{max}$ . Da  $\bar{\gamma}_{r_{max}} = \bar{\gamma}_{r_{max}}$  bereits nach Schritt 1 vorherrscht, ist in dieser Kompetenz keine Anpassung mehr notwendig. Wegen der Monotonieeigenschaft gilt jetzt immer noch  $\Delta\bar{z}_{r_{max}} < 0$ . Da  $\bar{\gamma}_{r_{max}} > \epsilon_0$  ist, kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein.

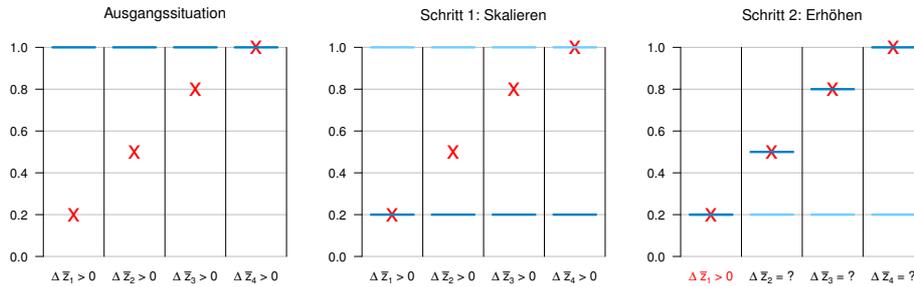


Abbildung 11: Veranschaulichung von Eindeutigkeitssatz A. Die  $\times$  markieren den beliebigen Vektor  $\bar{\gamma}$ , die  $\text{---}$  den aktuellen Zustand von  $\bar{\gamma}$ . **Bild 1:** Die Ausgangssituation. **Bild 2:** Da  $\bar{\gamma} = (1, 1, 1, 1)^T$  ein Fixpunkt ist, ändert sich an  $\Delta \bar{z} > 0$  nichts, wenn wir  $\bar{\gamma}$  skalieren. **Bild 3:** Nach der Erhöhung von  $\bar{\gamma}$  auf  $\bar{\gamma}$  können wir keine Aussage mehr über das Vorzeichen von  $\Delta \bar{z}_2$ ,  $\Delta \bar{z}_3$  und  $\Delta \bar{z}_4$  tätigen. Aus der Monotonieeigenschaft folgt aber, dass  $\Delta \bar{z}_1 > 0$  bleiben muss, folglich kann  $\bar{\gamma}$  kein Gleichgewicht sein.

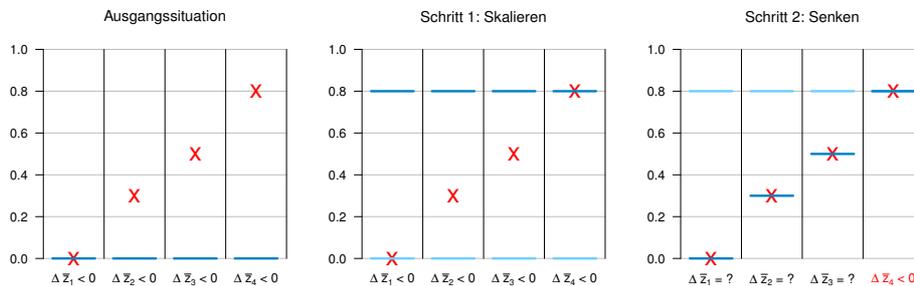


Abbildung 12: Veranschaulichung von Eindeutigkeitssatz C. Die  $\times$  markieren den beliebigen Vektor  $\bar{\gamma}$ , die  $\text{---}$  den aktuellen Zustand von  $\bar{\gamma}$ . **Bild 1:** Die Ausgangssituation. **Bild 2:** Da  $\bar{\gamma} = (\epsilon_0, \epsilon_0, \epsilon_0, \epsilon_0)^T$  ein Fixpunkt ist, ändert sich an  $\Delta \bar{z} < 0$  nichts, wenn wir  $\bar{\gamma}$  skalieren. **Bild 3:** Nach dem Senken von  $\bar{\gamma}$  auf  $\bar{\gamma}$  können wir keine Aussage mehr über das Vorzeichen von  $\Delta \bar{z}_1$ ,  $\Delta \bar{z}_2$  und  $\Delta \bar{z}_3$  tätigen. Aus der Monotonieeigenschaft folgt aber, dass  $\Delta \bar{z}_4 < 0$  bleiben muss, folglich kann  $\bar{\gamma}$  kein Gleichgewicht sein.

In Abbildung 12 auf der vorherigen Seite wird die Idee veranschaulicht.  $\square$

**Satz 36 (Eindeutigkeitssatz AC):** Sei  $\bar{\gamma}$  ein Gleichgewichtszustand, in dem für jede benötigte Kompetenz  $r \in \bar{\mathcal{R}}$  entweder Stabilität A ( $\bar{\gamma}_r = 1 \wedge \Delta\bar{z}_r > 0$ ) oder Stabilität C ( $\bar{\gamma}_r = \epsilon_0 \wedge \Delta\bar{z}_r < 0$ ) gilt. Dann ist dieser Gleichgewichtszustand eindeutig.

*Beweis.* Sei OBdA  $\bar{\mathcal{R}} = \mathcal{R}$ . Sei  $\bar{\mathcal{R}}_A = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = 1\}$  und  $\bar{\mathcal{R}}_C = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = \epsilon_0\}$ . Sei  $\bar{\gamma} \neq \bar{\gamma}$  ein beliebiger Vektor in  $[\epsilon_0, 1]^R$  und sei  $r_{min} = \arg \min_{r \in \bar{\mathcal{R}}_A} \bar{\gamma}_r$  bzw.  $r_{max} = \arg \max_{r \in \bar{\mathcal{R}}_C} \bar{\gamma}_r$ .

Wir unterscheiden zwei Fälle.

*Fall:*  $\bar{\gamma}_{r_{min}} < 1$

Gegeben, dass  $\bar{\gamma}$  ein Gleichgewicht ist, zeigen wir wiederum in zwei Schritten, dass  $\bar{\gamma}$  kein Gleichgewicht sein kann.

Schritt 1: Wir **skalieren** alle  $\bar{\gamma}_r$  um  $\bar{\gamma}_{r_{min}}$ . Dass dabei  $\bar{\gamma}_r = \epsilon_0 \cdot \bar{\gamma}_{r_{min}} < \epsilon_0$  für alle  $r \in \bar{\mathcal{R}}_C$  gilt, stört uns wegen Bemerkung 31 nicht weiter. Wegen der Skalierungsinvarianz bleibt  $\Delta\bar{z}$  unverändert. Es gilt also nach wie vor  $\Delta\bar{z}_r > 0$  für alle  $r \in \bar{\mathcal{R}}_A$ . Insbesondere herrscht  $\bar{\gamma}_r \leq \bar{\gamma}_r$  für alle  $r \in \bar{\mathcal{R}}$  vor.

Schritt 2: Wir **erhöhen**  $\bar{\gamma}_r$  auf  $\bar{\gamma}_r$  für alle  $r \neq r_{min}$ . Da  $\bar{\gamma}_{r_{min}} = \bar{\gamma}_{r_{min}}$  bereits nach Schritt 1 vorherrscht, ist in dieser Kompetenz keine Anpassung mehr notwendig. Wegen der Monotonieeigenschaft gilt jetzt immer noch  $\Delta\bar{z}_{r_{min}} > 0$ . Da  $\bar{\gamma}_{r_{min}} < 1$  ist, kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein.

Wir sammeln also die Erkenntnis, dass wir keinen weiteren Fixpunkt finden können, wenn wir Effizienzwerte der Menge  $\bar{\mathcal{R}}_A$  senken, egal, wie wir die Kompetenzen der Gruppe  $\bar{\mathcal{R}}_C$  handhaben. In Abbildung 13 auf der nächsten Seite wird die Situation veranschaulicht.

*Fall:*  $\bar{\gamma}_{r_{min}} = 1$

Das ist jener Fall, in welchem wir nur Kompetenzen der Gruppe  $\bar{\mathcal{R}}_C$  erhöhen und Kompetenzen der Gruppe  $\bar{\mathcal{R}}_A$  unangetastet lassen. Hier muss wegen  $\bar{\gamma} \neq \bar{\gamma}$  jedenfalls  $\bar{\gamma}_{r_{max}} > \epsilon_0$  gelten. Auch hier ist kein weiterer Fixpunkt auffindbar, wie wir in folgenden beiden Schritten erkennen:

Schritt 1: Wir **skalieren** alle  $\bar{\gamma}_r$  um  $\bar{\gamma}_{r_{max}}/\epsilon_0$ . Dass dabei  $\bar{\gamma}_r > 1$  für alle  $r \in \bar{\mathcal{R}}_A$  gilt, stört uns nicht weiter. Wegen der Skalierungsinvarianz bleibt  $\Delta\bar{z}$  unverändert. Es gilt also nach wie vor  $\Delta\bar{z}_r < 0$  für alle  $r \in \bar{\mathcal{R}}_C$ . Insbesondere herrscht  $\bar{\gamma}_r \geq \bar{\gamma}_r$  für alle  $r \in \bar{\mathcal{R}}$  vor.

Schritt 2: Wir **senken**  $\bar{\gamma}_r$  auf  $\bar{\gamma}_r$  für alle  $r \neq r_{max}$ . Da  $\bar{\gamma}_{r_{max}} = \bar{\gamma}_{r_{max}}$  bereits nach Schritt 1 vorherrscht, ist in dieser Kompetenz keine Anpassung mehr notwendig. Wegen der Monotonieeigenschaft gilt jetzt immer noch  $\Delta\bar{z}_{r_{max}} < 0$ . Da  $\bar{\gamma}_{r_{max}} > \epsilon_0$  ist, kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein.

In Abbildung 14 auf der nächsten Seite wird die Situation veranschaulicht.  $\square$

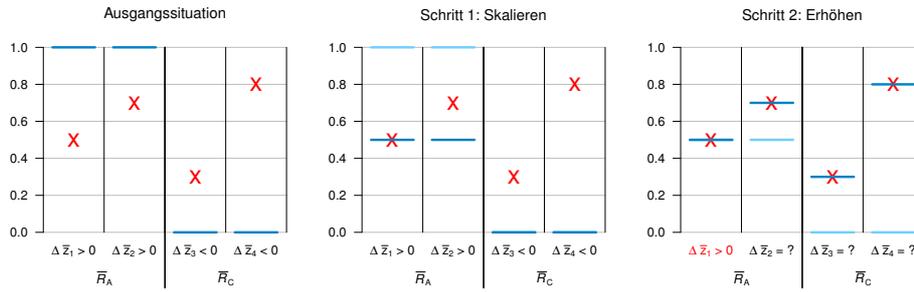


Abbildung 13: Veranschaulichung von Eindeutigkeitsatz AC ( $\bar{r}_{min} < 1$ ). Die **X** markieren  $\bar{\gamma}$ , die **—** den aktuellen Zustand von  $\bar{\gamma}$ . **Bild 1:** Die Ausgangssituation. **Bild 2:** Da  $\bar{\gamma} = (1, 1, \epsilon_0, \epsilon_0)^T$  ein Gleichgewicht ist, ändert sich an  $\Delta \bar{z}_1 > 0$  und  $\Delta \bar{z}_2 > 0$  nichts, wenn wir  $\bar{\gamma}$  skalieren. **Bild 3:** Nach dem Erhöhen von  $\bar{\gamma}$  auf  $\bar{\gamma}$  können wir keine Aussage mehr über das Vorzeichen von  $\Delta \bar{z}_2$ ,  $\Delta \bar{z}_3$  und  $\Delta \bar{z}_4$  tätigen. Aus der Monotonieeigenschaft folgt aber, dass  $\Delta \bar{z}_1 > 0$  bleiben muss, folglich kann  $\bar{\gamma}$  kein Gleichgewicht sein.

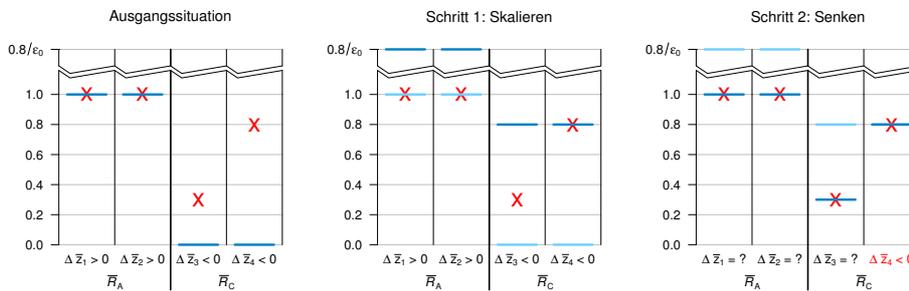


Abbildung 14: Veranschaulichung von Eindeutigkeitsatz AC ( $\bar{r}_{min} = 1$ ). Die **X** markieren  $\bar{\gamma}$ , die **—** den aktuellen Zustand von  $\bar{\gamma}$ . **Bild 1:** Die Ausgangssituation. **Bild 2:** Da  $\bar{\gamma} = (1, 1, \epsilon_0, \epsilon_0)^T$  ein Gleichgewicht ist, ändert sich an  $\Delta \bar{z}_3 < 0$  und  $\Delta \bar{z}_4 < 0$  nichts, wenn wir  $\bar{\gamma}$  skalieren. **Bild 3:** Nach dem Senken von  $\bar{\gamma}$  auf  $\bar{\gamma}$  können wir keine Aussage mehr über das Vorzeichen von  $\Delta \bar{z}_1$ ,  $\Delta \bar{z}_2$  und  $\Delta \bar{z}_3$  tätigen. Aus der Monotonieeigenschaft folgt aber, dass  $\Delta \bar{z}_4 < 0$  bleiben muss, folglich kann  $\bar{\gamma}$  kein Gleichgewicht sein.

Wir wissen also jetzt, dass jeder Fixpunkt, der am Rand liegt, eindeutig ist. Jetzt wenden wir uns den restlichen Fixpunkten zu. Ab jetzt benötigen wir die Voraussetzung, dass die Menge  $\bar{\mathcal{N}}(\bar{x})$  zusammenhängend ist.

**Lemma 37 (Eindeutigkeitslemma ABC):** *Sei die Menge  $\bar{\mathcal{N}}(\bar{x})$  zusammenhängend. Sei  $\bar{\gamma}$  ein Gleichgewichtszustand, in dem für mindestens eine benötigte Kompetenz  $r \in \bar{\mathcal{R}}$  Stabilität A ( $\bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0$ ) oder Stabilität C ( $\bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0$ ) und für mindestens eine benötigte Kompetenz Stabilität B ( $\bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0$ ) gilt.*

*Dann können wir keinen weiteren Gleichgewichtszustand auffinden, wenn wir lediglich Effizienzwerte jener Kompetenzen variieren, für die Stabilität B gilt.*

*Beweis.* Sei wie gehabt OBdA  $\bar{\mathcal{R}} = \mathcal{R}$ . Wir definieren folgende Mengen:

$$\begin{aligned}\bar{\mathcal{R}}_A &= \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0\} \\ \bar{\mathcal{R}}_C &= \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0\} \\ \bar{\mathcal{R}}_B &= \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0\}\end{aligned}$$

Sei  $\bar{\bar{\gamma}} \neq \bar{\gamma}$  ein beliebiger Vektor in  $[\epsilon_0, 1]^R$  mit der Einschränkung, dass

$$\bar{\bar{\gamma}}_r = \bar{\gamma}_r \quad \forall r \in \bar{\mathcal{R}}_A \cup \bar{\mathcal{R}}_C \quad (98)$$

*Fall:*  $\bar{\bar{\gamma}}_r \geq \bar{\gamma}_r \quad \forall r \in \bar{\mathcal{R}}_B$

Wir müssen zumindest ein  $\bar{\gamma}_r$  erhöhen und kein  $\bar{\gamma}_r$  senken, um zum Zustand  $\bar{\bar{\gamma}}$  zu gelangen. Seien  $r_{max} = \arg \max_{r \in \bar{\mathcal{R}}_B} \bar{\bar{\gamma}}_r / \bar{\gamma}_r$ ,  $q = \bar{\bar{\gamma}}_{r_{max}} / \bar{\gamma}_{r_{max}} > 1$  sowie  $\bar{\mathcal{R}}_B^{max} = \{r \in \bar{\mathcal{R}}_B : \bar{\bar{\gamma}}_r / \bar{\gamma}_r = q\}$ .

Klarerweise gilt  $\bar{\bar{\gamma}}_r > \epsilon_0$  für alle  $r \in \bar{\mathcal{R}}_B^{max}$ . Wir zeigen wie gehabt in zwei Schritten, dass  $\bar{\bar{\gamma}}$  kein Gleichgewicht sein kann.

Schritt 1: Wir **skalieren** alle  $\bar{\gamma}_r$  um  $q$ . Dass dabei  $\bar{\gamma}_r > 1$  für alle  $r \in \bar{\mathcal{R}}_A$  und möglicherweise für einige  $r \in \bar{\mathcal{R}}_B$  gilt, stört uns nicht weiter. Wegen der Skalierungsinvarianz bleibt  $\Delta \bar{z}$  unverändert. Es gilt also nach wie vor  $\Delta \bar{z}_r = 0$  für alle  $r \in \bar{\mathcal{R}}_B$  und insbesondere herrscht  $\bar{\gamma}_r \geq \bar{\bar{\gamma}}_r$  für alle  $r \in \bar{\mathcal{R}}$  vor.

Schritt 2: Wir **senken**  $\bar{\gamma}_\rho$  auf  $\bar{\bar{\gamma}}_\rho$  für alle  $\rho \notin \bar{\mathcal{R}}_B^{max}$ . Sei nun  $r \in \bar{\mathcal{R}}_B^{max}$  beliebig. Da  $\bar{\gamma}_r = \bar{\bar{\gamma}}_r$  bereits nach Schritt 1 vorherrscht, ist in dieser Kompetenz keine Anpassung mehr nötig. Wegen der Monotonieeigenschaft gilt jetzt zumindest  $\Delta \bar{z}_r \leq 0$ . Wir zeigen gleich, dass sogar  $\Delta \bar{z}_r < 0$  gelten muss und da  $\bar{\gamma}_r > \epsilon_0$  gilt, kann  $\bar{\bar{\gamma}}$  kein Gleichgewichtszustand sein.

Wir vollenden also Schritt 2, nachdem wir skaliert und gesenkt haben. Sei

$$\bar{\mathcal{R}}_B^* = \{r \in \bar{\mathcal{R}}_B^{max} : \Delta \bar{z}_r = 0\} \subseteq \bar{\mathcal{R}}_B^{max} \quad (99)$$

Angenommen,  $\bar{\mathcal{R}}_B^*$  ist nicht leer. Wir werfen nochmals einen Blick auf  $\Delta \bar{z}_r$ :

$$\Delta \bar{z}_r = -\beta_r + \alpha_r \sum_{i \in \mathcal{N}} \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \bar{x}_i$$

In Schritt 2 hat sich  $\bar{\gamma}_r$  für alle  $r \in \bar{\mathcal{R}}_B^*$  nicht verändert, da  $\bar{\gamma}_r = \bar{\bar{\gamma}}_r$  bereits erfüllt war. Da nach wie vor  $\Delta\bar{z}_r = 0$  für alle  $r \in \bar{\mathcal{R}}_B^*$  gilt, können sich in Schritt 2 auch sämtliche  $\bar{\eta}_i$  nicht verändert haben, welche Kompetenzen der Menge  $\bar{\mathcal{R}}_B^*$  benötigen!

Das bedeutet, wir können die Menge aller zu bearbeitenden Projektklassen  $\bar{\mathcal{N}}$  derart in zwei nichtleere Mengen  $\bar{\mathcal{N}}^*$  und  $\bar{\mathcal{N}}/\bar{\mathcal{N}}^*$  aufteilen, sodass

$$\bigcup_{i \in \bar{\mathcal{N}}^*} \mathcal{R}_i = \bar{\mathcal{R}}_B^* \quad \bigcup_{i \in \bar{\mathcal{N}}/\bar{\mathcal{N}}^*} \mathcal{R}_i = \bar{\mathcal{R}}/\bar{\mathcal{R}}_B^*$$

Das ist ein Widerspruch zu unserer Voraussetzung, dass  $\bar{\mathcal{N}}$  zusammenhängend ist. Folglich muss die Menge  $\bar{\mathcal{R}}_B^*$  leer sein und somit gilt  $\Delta\bar{z}_r < 0$  für alle  $r \in \bar{\mathcal{R}}_B^{max}$ , womit die Lücke in Schritt 2 geschlossen ist. In Abbildung 15 wird die Situation veranschaulicht.

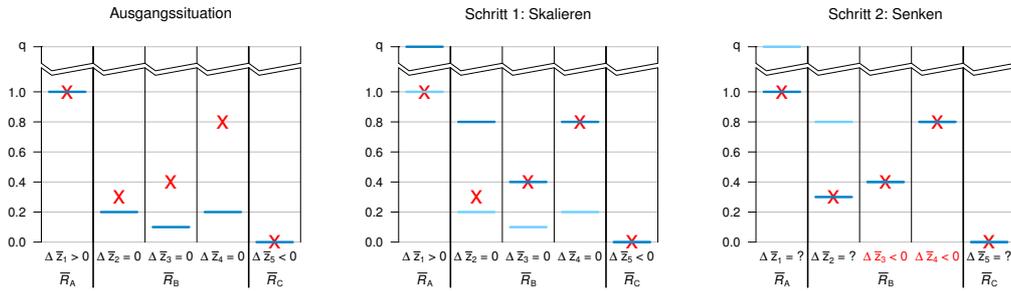


Abbildung 15: Veranschaulichung von Eindeutigkeitslemma ABC ( $\bar{\bar{\gamma}}_r \geq \bar{\gamma}_r \forall r \in \bar{\mathcal{R}}_B$ ). Die **X** markieren  $\bar{\gamma}_r$ , die **—** den aktuellen Zustand von  $\bar{\gamma}_r$ . **Bild 1:** Die Ausgangssituation. **Bild 2:** Da  $\bar{\gamma} = (1, 0.2, 0.1, 0.2, \epsilon_0)^T$  ein Gleichgewicht ist, ändert sich an  $\Delta\bar{z}_2 = 0$ ,  $\Delta\bar{z}_3 = 0$  und  $\Delta\bar{z}_4 = 0$  nichts, wenn wir  $\bar{\gamma}$  skalieren. Dabei ist der Skalierungsfaktor  $q = \max(0.3/0.2, 0.4/0.1, 0.8/0.2) = 4$ . **Bild 3:** Nach dem Senken von  $\bar{\gamma}$  auf  $\bar{\bar{\gamma}}$  können wir keine Aussage mehr über das Vorzeichen von  $\Delta\bar{z}_1$ ,  $\Delta\bar{z}_2$  und  $\Delta\bar{z}_5$  tätigen. Aus der Monotonieeigenschaft folgt jedoch, dass  $\Delta\bar{z}_3 \leq 0$  und  $\Delta\bar{z}_4 \leq 0$  gelten muss. Da, wie wir aber gezeigt haben, sogar  $\Delta\bar{z}_3 < 0$  und  $\Delta\bar{z}_4 < 0$  gilt, kann  $\bar{\bar{\gamma}}$  kein Gleichgewicht sein.

*Fall:*  $\exists r \in \bar{\mathcal{R}}_B : \bar{\bar{\gamma}}_r < \bar{\gamma}_r$

In diesem Fall müssen wir zumindest ein  $\bar{\gamma}_r$  senken, um zum Zustand  $\bar{\bar{\gamma}}$  zu gelangen. Seien  $r_{min} = \arg \min_{r \in \bar{\mathcal{R}}_B} \bar{\bar{\gamma}}_r / \bar{\gamma}_r$ ,  $q = \bar{\bar{\gamma}}_{r_{min}} / \bar{\gamma}_{r_{min}} < 1$  sowie  $\bar{\mathcal{R}}_B^{min} = \{r \in \bar{\mathcal{R}}_B : \bar{\bar{\gamma}}_r / \bar{\gamma}_r = q\}$ .

Wir zeigen wie gehabt in zwei Schritten, dass  $\bar{\bar{\gamma}}$  kein Gleichgewicht sein kann.

**Schritt 1:** Wir **skalieren** alle  $\bar{\gamma}_r$  um  $q$ . Dass dabei  $\bar{\gamma}_r < \epsilon_0$  für alle  $r \in \bar{\mathcal{R}}_C$  und möglicherweise für einige  $r \in \bar{\mathcal{R}}_B$  gilt, stört uns nicht weiter. Wegen der Skalierungsinvarianz bleibt  $\Delta\bar{z}$  unverändert. Es gilt also nach wie vor  $\Delta\bar{z}_r = 0$  für alle  $r \in \bar{\mathcal{R}}_B$  und insbesondere herrscht  $\bar{\gamma}_r \leq \bar{\bar{\gamma}}_r$  für alle  $r \in \bar{\mathcal{R}}$  vor.

Schritt 2: Wir **erhöhen**  $\bar{\gamma}_r$  auf  $\bar{\gamma}_r$  für alle  $\rho \notin \bar{\mathcal{R}}_B^{min}$ . Sei nun  $r \in \bar{\mathcal{R}}_B^{min}$  beliebig. Da  $\bar{\gamma}_r = \bar{\gamma}_r$  bereits nach Schritt 1 vorherrscht, ist in dieser Kompetenz keine Anpassung mehr nötig. Wegen der Monotonieeigenschaft gilt jetzt zumindest  $\Delta \bar{z}_r \geq 0$ . Aus dem gleichen Grund wie im vorherigen Fall gilt sogar  $\Delta \bar{z}_r > 0$  und da  $\bar{\gamma}_r < 1$  gilt, kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein.

In Abbildung 16 wird die Situation veranschaulicht. □

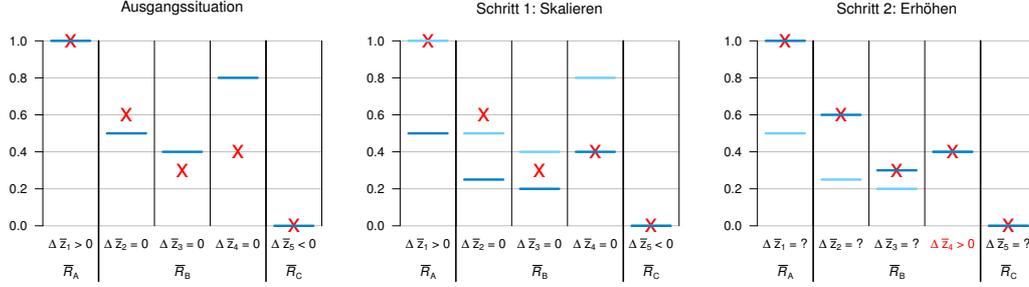


Abbildung 16: Veranschaulichung von Eindeutigkeitslemma ABC ( $\exists r \in \bar{\mathcal{R}}_B : \bar{\gamma}_r < \bar{\gamma}_r$ ). Die **X** markieren  $\bar{\gamma}$ , die **—** den aktuellen Zustand von  $\bar{\gamma}$ . **Bild 1:** Die Ausgangssituation. **Bild 2:** Da  $\bar{\gamma} = (1, 0.5, 0.4, 0.8, \epsilon_0)^T$  ein Gleichgewicht ist, ändert sich an  $\Delta \bar{z}_2 = 0$ ,  $\Delta \bar{z}_3 = 0$  und  $\Delta \bar{z}_4 = 0$  nichts, wenn wir  $\bar{\gamma}$  skalieren. Der Skalierungsfaktor ist  $q = \min(0.6/0.5, 0.3/0.4, 0.4/0.8) = 1/2$ . **Bild 3:** Nach dem Erhöhen von  $\bar{\gamma}$  auf  $\bar{\gamma}$  können wir keine Aussage mehr über das Vorzeichen von  $\Delta \bar{z}_1$ ,  $\Delta \bar{z}_2$ ,  $\Delta \bar{z}_3$  und  $\Delta \bar{z}_5$  tätigen. Aus der Monotonieeigenschaft folgt jedoch, dass  $\Delta \bar{z}_4 \geq 0$  gelten muss. Da, wie wir aber überlegt haben, sogar  $\Delta \bar{z}_4 > 0$  gilt, kann  $\bar{\gamma}$  kein Gleichgewicht sein.

Falls  $\bar{\mathcal{N}}(\bar{x})$  nicht zusammenhängend ist, so sind die Aussagen von Lemma 37 nicht gültig.

**Beispiel 38:** Ein simples Beispiel mit  $n = 2$ ,  $R = 3$  und  $\sigma = 1$ .

$$d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/3 & 2/3 \end{pmatrix} \quad \beta = \begin{pmatrix} 1/8 \\ 1/6 \\ 1/3 \end{pmatrix} \quad \alpha = \begin{pmatrix} 5/8 \\ 1/2 \\ 5/4 \end{pmatrix} \quad \bar{x} = \begin{pmatrix} 2/5 \\ 3/5 \end{pmatrix}$$

Wie wir leicht erkennen können, ist  $\bar{\mathcal{N}}(\bar{x})$  nicht zusammenhängend, da die Projektklassen 1 und 2 keine gemeinsame Kompetenz erfordern. Ergo würde uns Algorithmus 5 die Liste  $\mathcal{M} = \{\{1\}, \{2\}\}$  zurückgeben. Für den Vektor

$$\bar{\gamma} = (1, 2/5, 1)^T$$

bestimmen wir die beiden Effizienzwerte für die Projektklassen

$$\bar{\eta}_1 = \frac{1}{\sum_{r \in \mathcal{R}_1} \frac{d_{1r}}{\bar{\gamma}_r}} = \frac{1}{1} = 1 \quad \text{und} \quad \bar{\eta}_2 = \frac{1}{\sum_{r \in \mathcal{R}_2} \frac{d_{2r}}{\bar{\gamma}_r}} = \frac{1}{\frac{1/3}{2/5} + \frac{2/3}{1}} = 2/3$$

und kontrollieren an Hand von  $\Delta\bar{z}$ , ob wir ein Gleichgewicht haben:

$$\Delta\bar{z}_1 = -\beta_1 + \alpha_1 \sum_{i=1}^2 \frac{d_{i1}}{\bar{\gamma}_1} \bar{\eta}_i \bar{x}_i = -1/8 + 5/8 \cdot \left( \frac{1}{1} \cdot 1 \cdot 2/5 + \frac{0}{1} \cdot 2/3 \cdot 3/5 \right) = 1/8 > 0 \quad \checkmark$$

$$\Delta\bar{z}_2 = -\beta_2 + \alpha_2 \sum_{i=1}^2 \frac{d_{i2}}{\bar{\gamma}_2} \bar{\eta}_i \bar{x}_i = -1/6 + 1/2 \cdot \left( \frac{0}{2/5} \cdot 1 \cdot 2/5 + \frac{1/3}{2/5} \cdot 2/3 \cdot 3/5 \right) = 0 \quad \checkmark$$

$$\Delta\bar{z}_3 = -\beta_3 + \alpha_3 \sum_{i=1}^2 \frac{d_{i3}}{\bar{\gamma}_3} \bar{\eta}_i \bar{x}_i = -1/3 + 5/4 \cdot \left( \frac{0}{1} \cdot 1 \cdot 2/5 + \frac{2/3}{1} \cdot 2/3 \cdot 3/5 \right) = 0 \quad \checkmark$$

Für Projektklasse 1 gilt also Stabilität A, für die Projektklassen 2 und 3 Stabilität B. Weil  $\bar{N}(\bar{x})$  nicht zusammenhängend ist, können wir an  $\bar{\gamma}_2$  und  $\bar{\gamma}_3$  schrauben, ohne die 1. Kompetenz zu beeinflussen. Man kann leicht nachrechnen, dass jeder Vektor der Form

$$\bar{\gamma} = (1, k \cdot 2/5, k \cdot 1)^T$$

für  $5/2 \cdot \epsilon_0 \leq k \leq 1$  ein gültiges Gleichgewicht ist. Keine Spur von Eindeutigkeit.

**Satz 39 (Eindeutigkeitssatz ABC):** Sei die Menge  $\bar{N}(\bar{x})$  zusammenhängend. Sei der Vektor  $\bar{\gamma}$  ein Fixpunkt, in dem für mindestens eine benötigte Kompetenz  $r \in \bar{\mathcal{R}}$  Stabilität A ( $\bar{\gamma}_r = 1 \wedge \Delta\bar{z}_r > 0$ ) oder Stabilität C ( $\bar{\gamma}_r = \epsilon_0 \wedge \Delta\bar{z}_r < 0$ ) und für mindestens eine benötigte Kompetenz Stabilität B ( $\bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta\bar{z}_r = 0$ ) gilt. Dann ist dieser Gleichgewichtszustand eindeutig.

*Beweis.* Der Beweis des allgemeinen Falles ist durch unsere Vorarbeiten sehr schnell erledigt! Sei OBdA  $\bar{\mathcal{R}} = \mathcal{R}$ . Wir definieren folgende Mengen:

$$\begin{aligned} \bar{\mathcal{R}}_A &= \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = 1 \wedge \Delta\bar{z}_r > 0\} \\ \bar{\mathcal{R}}_C &= \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = \epsilon_0 \wedge \Delta\bar{z}_r < 0\} \\ \bar{\mathcal{R}}_B &= \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta\bar{z}_r = 0\} \end{aligned}$$

Eindeutigkeitslemma ABC besagt, dass es unmöglich ist, durch bloßes Verschieben der Effizienzwerte der Kompetenzen in  $\bar{\mathcal{R}}_B$  einen weiteren Fixpunkt zu finden. Es ist auch nicht möglich, einen weiteren Fixpunkt zu finden, wenn wir ausschließlich Effizienzwerte der Kompetenzen in  $\bar{\mathcal{R}}_A$  und/oder in  $\bar{\mathcal{R}}_C$  variieren. Hier verweisen wir auf die Beweise der Eindeutigkeitssätze A und C.

Die restlichen drei Möglichkeiten behandeln wir in drei Teilen.

**Teil AB:** Sofern  $\bar{\mathcal{R}}_A \neq \emptyset$ : Nur Effizienzwerte der Mengen  $\bar{\mathcal{R}}_A$  und  $\bar{\mathcal{R}}_B$  variieren.

Sei  $\bar{\gamma} \neq \bar{\gamma}$  ein beliebiger Vektor in  $[\epsilon_0, 1]^R$  mit der Einschränkung, dass

$$\begin{aligned} \bar{\gamma}_r &\neq \bar{\gamma}_r \quad \text{für mind. ein } r \in \bar{\mathcal{R}}_A \text{ und} \\ \bar{\gamma}_r &\neq \bar{\gamma}_r \quad \text{für mind. ein } r \in \bar{\mathcal{R}}_B \end{aligned}$$

Seien  $r_{min} = \arg \min_{r \in \bar{\mathcal{R}}} \bar{\gamma}_r / \gamma_r$  sowie  $q = \bar{\gamma}_{r_{min}} / \gamma_{r_{min}} < 1$ .

*Fall:*  $\exists r \in \bar{\mathcal{R}}_A : \bar{\gamma}_r / \gamma_r = q$

Wir verweisen auf die Beweise der Eindeutigkeitssätze A und AC, in welchen alle benötigten Argumente dargelegt werden, um diesen Fall zu behandeln. Nach Anwendung der gewohnten beiden Schritte **Skalieren** und **Erhöhen** gilt: Es gibt ein  $r \in \bar{\mathcal{R}}_A$ , sodass  $\bar{\gamma}_r < 1$  und  $\Delta \bar{z}_r > 0$ . Folglich kann  $\bar{\gamma}$  kein Gleichgewicht sein.

*Fall:*  $\nexists r \in \bar{\mathcal{R}}_A : \bar{\gamma}_r / \gamma_r = q$

Auch in diesem Fall kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein. Die Beweisführung gleicht jener von Eindeutigkeitslemma ABC (Fall:  $\exists r \in \bar{\mathcal{R}}_B : \bar{\gamma}_r < \gamma_r$ ).

**Teil BC:** Sofern  $\bar{\mathcal{R}}_C \neq \emptyset$ : Nur Effizienzwerte der Mengen  $\bar{\mathcal{R}}_B$  und  $\bar{\mathcal{R}}_C$  variieren.

Sei  $\bar{\gamma} \neq \gamma$  ein beliebiger Vektor in  $[\epsilon_0, 1]^R$  mit der Einschränkung, dass

$$\begin{aligned} \bar{\gamma}_r &\neq \gamma_r && \text{für mind. ein } r \in \bar{\mathcal{R}}_B \text{ und} \\ \bar{\gamma}_r &\neq \gamma_r && \text{für mind. ein } r \in \bar{\mathcal{R}}_C \end{aligned}$$

Seien  $r_{max} = \arg \max_{r \in \bar{\mathcal{R}}} \bar{\gamma}_r / \gamma_r$  sowie  $q = \bar{\gamma}_{r_{max}} / \gamma_{r_{max}} > 1$ .

*Fall:*  $\exists r \in \bar{\mathcal{R}}_C : \bar{\gamma}_r / \gamma_r = q$

Wir verweisen auf die Beweise der Eindeutigkeitssätze C und AC, in welchen alle benötigten Argumente dargelegt werden, um diesen Fall zu behandeln. Nach Anwendung der gewohnten beiden Schritte **Skalieren** und **Senken** gilt: Es gibt ein  $r \in \bar{\mathcal{R}}_C$ , sodass  $\bar{\gamma}_r > \epsilon_0$  und  $\Delta \bar{z}_r < 0$ . Folglich kann  $\bar{\gamma}$  kein Gleichgewicht sein.

*Fall:*  $\nexists r \in \bar{\mathcal{R}}_C : \bar{\gamma}_r / \gamma_r = q$

Auch in diesem Fall kann  $\bar{\gamma}$  kein Gleichgewichtszustand sein. Die Beweisführung gleicht jener von Eindeutigkeitslemma ABC.

**Teil ABC:** Sofern  $\bar{\mathcal{R}}_A \neq \emptyset$  und  $\bar{\mathcal{R}}_C \neq \emptyset$ : Mindestens einen Effizienzwert in jeder der drei Mengen variieren.

Wir stellen fest, dass sich dieser Teil auf Teil AB oder Teil BC zurückführen lässt und daher auch hier kein weiterer Fixpunkt auffindbar ist.

Zuletzt bemerken wir noch, dass es in Teil AB völlig unerheblich ist, ob die Menge  $\bar{\mathcal{R}}_C$  leer ist oder nicht. Gleiches gilt in Teil BC mit der Menge  $\bar{\mathcal{R}}_A$ . Unterm Strich ist damit die Eindeutigkeit des Fixpunktes gezeigt.  $\square$

**Satz 40 (Eindeutigkeitssatz B):** Sei die Menge  $\bar{N}(\bar{x})$  zusammenhängend. Sei  $\bar{\gamma}$  ein Fixpunkt, in dem für alle benötigten Kompetenzen Stabilität B ( $\bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0$ ) gilt. Dann folgt:

1. Jeder Vektor  $\bar{\gamma}'$  der Form

$$\bar{\gamma}'_r = \begin{cases} \bar{\gamma}_r & \text{falls } r \in \mathcal{R}/\bar{\mathcal{R}} \\ k \cdot \bar{\gamma}_r & \text{falls } r \in \bar{\mathcal{R}} \end{cases}$$

ist für geeignetes  $k > 0$  (also so, dass  $\bar{\gamma}' \in [\epsilon_0, 1]^R$  gilt) ebenfalls ein Gleichgewicht.

2. Die auf diese Art gewonnene überabzählbar unendlich große Menge an Gleichgewichtszuständen sind die einzig möglichen.

*Beweis.* Sei OBdA  $\bar{\mathcal{R}} = \mathcal{R}$ . Dass jeder Vektor  $\bar{\gamma}'$  für geeignetes  $k > 0$  ebenfalls ein Fixpunkt ist, dürfte offensichtlich sein – Stichwort Skalierungsinvarianz.

Um Punkt 2 zu beweisen, verweisen wir auf den Beweis von Eindeutigkeitslemma ABC. Wir müssen dort bei der Wahl von  $\bar{\gamma}$  lediglich Bedingung (98) durch folgende ersetzen:

$$\frac{\bar{\gamma}_r}{\bar{\gamma}_r} \neq \frac{\bar{\gamma}_\rho}{\bar{\gamma}_\rho} \quad \text{für ein } r \neq \rho$$

Den restlichen Beweis könnten wir praktisch 1 zu 1 abschreiben. □

Falls für ein Grenzportfolio  $\bar{x}$  die Menge  $\bar{N}(\bar{x})$  nicht zusammenhängend ist, so können wir sie mit Hilfe von Algorithmus 5 in alle minimalen Inselbetriebe zerlegen. Lässt sich jetzt auf keinen dieser Inselbetriebe Eindeutigkeitssatz B anwenden, so ist der Fixpunkt eindeutig, andernfalls nicht.

### 5.2.6. Konvergenz zum Gleichgewichtszustand

Wir wissen also, dass es für ein Grenzportfolio  $\bar{x}$  einen Gleichgewichtszustand für das Kompetenzentwicklungssystem gibt. Doch wie sieht es mit der Konvergenz zu einem Fixpunkt aus, wenn wir in jeder Periode das selbe Grenzportfolio  $\bar{x}$  anwenden? Diese Frage bleibt in dieser Arbeit offen.

Wir können nur sagen, dass Fixpunkte per se keine Grenzwerte im mathematischen Sinn darstellen. Wenn allerdings das Kompetenzentwicklungssystem konvergiert, dann gegen einen Fixpunkt. Fixpunkte sind also die einzigen Kandidaten für Grenzwerte.

**Beispiel 41 (Oszillationsbeispiel):** Eine naheliegende und einfache Möglichkeit, etwas über das Grenzverhalten des Kompetenzentwicklungssystems zu erfahren, ist dieses System durchzuitieren. Wir geben Startwerte  $z_{r1} \in \mathbb{R}$  vor und setzen die aktuellen Kompetenzscores immer wieder aufs Neue in (83)-(84) ein und beobachten die Entwicklung.

$r$	Kompetenz	$d_{ir}$	$\beta_r$	$\alpha_r$
1	Schwimmen	0.45	0.1	0.7
2	Radfahren	0.35	0.2	0.4
3	Laufen	0.20	0.3	0.9

Tabelle 5: Die Projektklasse Triathlon: Daten für das Szenario im Oszillationsbeispiel

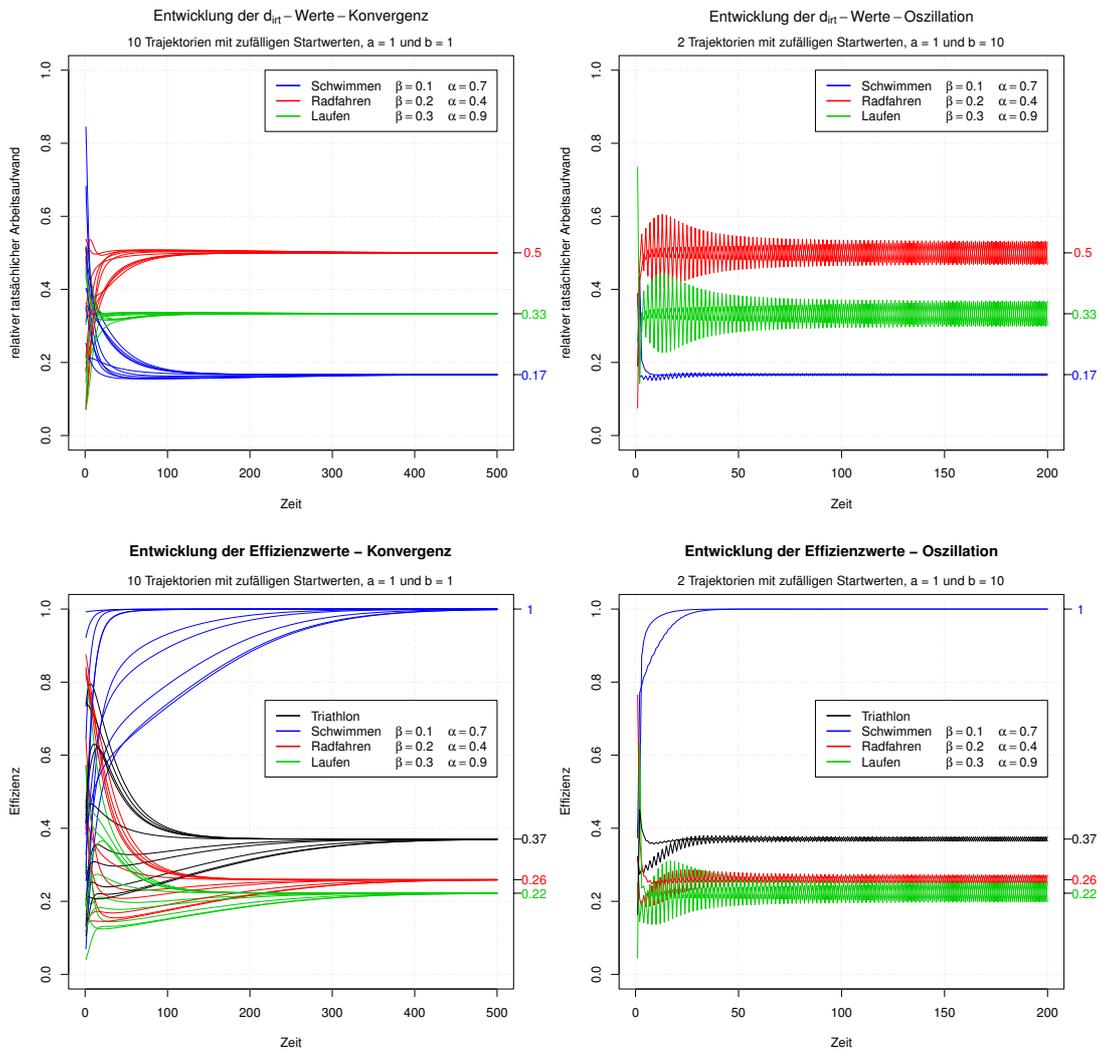


Abbildung 17: Entwicklung von  $d_{irt}$  und  $\gamma_{rt}$  im Oszillationsbeispiel. **Links:** Die Werte für  $d_{irt}$  (oben) und  $\gamma_{rt}$  (unten) streben für  $b=1$  gegen den eindeutigen Fixpunkt. **Rechts:** Hier setzen wir  $b=10$ . Um die Übersicht zu erhöhen, betrachten wir lediglich 2 Trajektorien und nur 200 Perioden. Oszillation um den Fixpunkt ist die Folge.

Wir werden dies anhand des Triathlon-Beispiels tun. Tabelle 5 zeigt die Daten für unser Szenario, das wir betrachten und Abbildung 17 auf der vorherigen Seite zeigt die Entwicklung von  $d_{it}$  und  $\gamma_{rt}$  für zwei verschiedene Werte für  $b$ .

Wir stellen die Vermutung auf, dass das Kompetenzentwicklungssystem langfristig zu einem Fixpunkt tendiert. Aber selbst wenn diese Vermutung zutrifft, so wird ein Fixpunkt möglicherweise nie (genau) erreicht. Wie nahe das Kompetenzentwicklungssystem einem Fixpunkt tatsächlich kommt, hängt sodann von der Feinheit des Systems ab, welche im Wesentlichen durch die Parameter  $\alpha_r$ ,  $\beta_r$  und  $b$  bestimmt wird. Hohe Werte eben genannter Parameter können zu einer Oszillation des Systems um einen Gleichgewichtszustand oder zu einem anderen chaotischen Verhalten führen.

Die eben genannten Parameter sind dann zu groß, wenn die Periodenlänge zu grob gewählt wurde. Das Leben, das auch die Bearbeitung von Projekten diverser Projektklassen umfasst, ist stetig und wird in unserem Modell diskretisiert. Nun ist es aus diversen Gründen natürlich nicht sinnvoll, jede Millisekunde zu simulieren, aber die Periodenlänge sollte zumindest so klein gewählt werden, dass es zu keiner erkennbaren Oszillation kommt.

### 5.2.7. Grenzgewinn und Lösungsliste

Sobald wir wissen, wie die Effizienzwerte  $\bar{\eta}_i$  im Gleichgewicht aussehen, können wir den Grenzgewinn des Projektportfolios berechnen.

**Definition 42:** *Unter dem **Grenzgewinn** des Grenzportfolios  $\bar{x}$ , den wir mit  $\bar{w}(\bar{x})$  bezeichnen, verstehen wir jenen Gewinn, den wir in einer Periode unbeachtet allfälliger Abzinsungen erzielen können, wenn wir das Grenzportfolio  $\bar{x}$  anwenden und sich das Kompetenzentwicklungssystem in einem Gleichgewicht befindet, also*

$$\bar{w}(\bar{x}) = \sum_{i=1}^n w_i \bar{\eta}_i \bar{x}_i \quad (100)$$

Dabei skalieren wir  $\bar{\gamma}$  derart, dass für jeden Inselbetrieb  $\bar{N}_{IB}$ , in dem für alle in ihm benötigten Kompetenzen  $\bar{\mathcal{R}}_{IB}$  Stabilität  $B$  gilt, folgendes erfüllt ist:

$$\max_{r \in \bar{\mathcal{R}}_{IB}} \bar{\gamma}_r = 1 \quad (101)$$

Oftmals schreiben wir auch einfach  $\bar{w}$ , wenn aus dem Zusammenhang klar hervorgeht, um welches Grenzportfolio es sich handelt.

Jetzt haben wir alle Vorbereitungen abgeschlossen und können uns (endlich) der Berechnung der Fixpunkte widmen. Unsere Prozedur zur Berechnung des Gleichgewichtszustandes verwendet folgende Liste, um die Lösung zu speichern:

$$L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w}) \quad (102)$$

Wir greifen also im Folgenden stets auf eine Liste zurück, welche uns das zugrunde liegende Grenzportpolio  $\bar{x}$ , die Vektoren mit den dazu passenden (gegebenenfalls skalierten) Effizienzwerten für die Kompetenzen  $\bar{\gamma}$  und die Projektklassen  $\bar{\eta}$  im Gleichgewicht sowie den entsprechenden Grenzgewinn des Grenzportfolios  $\bar{w}$  liefert.

### 5.2.8. Zusammenfassung und Beispiel

Wir fassen die wichtigsten Resultate von Abschnitt 5.2 zusammen:

- **Gleichgewichte:** Der Vektor  $\bar{\gamma} \in [\epsilon_0, 1]^R$  ist genau dann ein Fixpunkt, wenn auf jede Kompetenz  $r \in \mathcal{R}$  genau eine der folgenden drei Möglichkeiten zutrifft:

Stabilität A:  $\bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0$

Stabilität B:  $\bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0$

Stabilität C:  $\bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0$

- **Skalierungsinvarianz:**  $\Delta \bar{z}$  ist invariant gegenüber Skalierungen von  $\bar{\gamma}$ .
- **Monotonieeigenschaft:**  $\Delta \bar{z}_r$  ist monoton fallend in  $\bar{\gamma}_r$  und monoton wachsend in  $\bar{\gamma}_\rho$  für  $\rho \neq r$ .
- **Existenz:** Gleichgewichtszustände für  $\bar{\gamma}$  existieren immer.
- **Eindeutigkeit:** Zu einem gegebenen Grenzportfolio  $\bar{x}$  ist der Gleichgewichtszustand  $\bar{\gamma}$  genau dann eindeutig, wenn es in jedem Inselbetrieb mindestens eine Kompetenz gibt, für die Stabilität A oder Stabilität C gilt.
- **Konvergenz:** Konvergenz zum Fixpunkt kann in dieser Arbeit nicht gezeigt werden, allerdings vermuten wir eine Tendenz zum Fixpunkt.

**Beispiel 43 (Stabilitätsbeispiel):** Wir werden das alt bekannte Triathlonbeispiel heranziehen, um die Ideen von Abschnitt 5.2 zu veranschaulichen. Tabelle 6 zeigt die Daten für jene vier Szenarien, welche wir betrachten werden. Wir setzen  $\sigma = 1$  und  $w = 1$ , der Grenzgewinn beträgt also in jedem Szenario  $\bar{\eta}$ .

$r$	Kompetenz	$d_r$	Stab. A		Stab. AB		Stab. B		Stab. BC	
			$\beta_r$	$\alpha_r$	$\beta_r$	$\alpha_r$	$\beta_r$	$\alpha_r$	$\beta_r$	$\alpha_r$
1	Schwimmen	0.45	0.3	0.7	0.1	0.7	0.1	0.6	0.1	0.5
2	Radfahren	0.35	0.1	0.4	0.2	0.4	0.2	0.4	0.2	0.4
3	Laufen	0.20	0.1	0.9	0.3	0.9	0.3	0.9	0.3	0.9

Tabelle 6: Die Projektklasse Triathlon: Szenarien für das Stabilitätsbeispiel

Wir gehen die vier Szenarien der Reihe nach durch.

### Stabilität A

Wir setzen alle  $\bar{\gamma}_r = 1$  und bestimmen  $\Delta\bar{z}$ :

$$\Delta\bar{z}_1 = -0.3 + 0.7 \cdot 0.45 = 0.015 > 0 \quad \checkmark$$

$$\Delta\bar{z}_2 = -0.1 + 0.4 \cdot 0.35 = 0.04 > 0 \quad \checkmark$$

$$\Delta\bar{z}_3 = -0.1 + 0.9 \cdot 0.20 = 0.08 > 0 \quad \checkmark$$

Alle Maximalgeschwindigkeiten können gehalten werden, also trifft auf alle Kompetenzen Stabilität A zu und  $\bar{\gamma} = (1, 1, 1)^T$  ist gemäß Eindeutigkeitsatz A der eindeutige Fixpunkt.

### Stabilität AB

Wieder setzen wir alle  $\bar{\gamma}_r = 1$  und bestimmen  $\Delta\bar{z}$ :

$$\Delta\bar{z}_1 = -0.1 + 0.7 \cdot 0.45 = 0.215 > 0 \quad \checkmark$$

$$\Delta\bar{z}_2 = -0.2 + 0.4 \cdot 0.35 = -0.06 < 0 \quad \times$$

$$\Delta\bar{z}_3 = -0.3 + 0.9 \cdot 0.20 = -0.12 < 0 \quad \times$$

Die Maximalgeschwindigkeiten beim Radfahren und Laufen können nicht gehalten werden, daher wissen wir, dass  $\bar{\gamma}_2 < 1$  und  $\bar{\gamma}_3 < 1$  zutreffen muss. Mit Hilfe einer etwas längeren Rechnung kann hier der gesuchte Vektor  $\bar{\gamma}$  sogar analytisch bestimmt werden.

$$\bar{\gamma} = (1, 7/27, 2/9)^T$$

Daraus ergibt sich die mittlere Projektgeschwindigkeit von

$$\bar{\eta} = \left( \sum_{r \in \bar{\mathcal{R}}} \frac{d_r}{\bar{\gamma}_r} \right)^{-1} = \left( \frac{0.45}{1} + \frac{0.35}{7/27} + \frac{0.2}{2/9} \right)^{-1} = \left( \frac{315 + 945 + 630}{700} \right)^{-1} = \frac{70}{189} \approx 0.37$$

Wir berechnen wieder  $\Delta\bar{z}$ :

$$\Delta\bar{z}_1 = -0.1 + 0.7 \cdot 0.45 \cdot \frac{70/189}{1} = \frac{1}{60} > 0 \quad \checkmark$$

$$\Delta\bar{z}_2 = -0.2 + 0.4 \cdot 0.35 \cdot \frac{70/189}{7/27} = 0 \quad \checkmark$$

$$\Delta\bar{z}_3 = -0.3 + 0.9 \cdot 0.20 \cdot \frac{70/189}{2/9} = 0 \quad \checkmark$$

Gemäß Definition 27 erfüllt der Vektor  $\bar{\gamma} = (1, 7/27, 2/9)^T$  alle Voraussetzungen um ein Fixpunkt zu sein und ist nach Eindeutigkeitsatz ABC auch eindeutig.

### Stabilität B

Wir stellen fest, dass

$$\bar{\gamma} = (1, 7/27, 2/9)^T$$

und für die dazugehörige mittlere Projekteffizienz

$$\bar{\eta} = \left( \sum_{r \in \bar{\mathcal{R}}} \frac{d_r}{\bar{\gamma}_r} \right)^{-1} = \left( \frac{0.45}{1} + \frac{0.35}{7/27} + \frac{0.2}{2/9} \right)^{-1} = \left( \frac{315 + 945 + 630}{700} \right)^{-1} = \frac{70}{189} \approx 0.37$$

ein Gleichgewichtszustand ist. Wir bestimmen den Vektor  $\Delta \bar{z}$ :

$$\Delta \bar{z}_1 = -0.1 + 0.6 \cdot 0.45 \cdot \frac{70/189}{1} = 0 \quad \checkmark$$

$$\Delta \bar{z}_2 = -0.2 + 0.4 \cdot 0.35 \cdot \frac{70/189}{7/27} = 0 \quad \checkmark$$

$$\Delta \bar{z}_3 = -0.3 + 0.9 \cdot 0.20 \cdot \frac{70/189}{2/9} = 0 \quad \checkmark$$

Also gilt für alle benötigten Kompetenzen Stabilität B und Eindeutigkeitssatz B sagt uns, dass jeder beliebige Vektor

$$(\bar{\gamma}_1, \bar{\gamma}_2, \bar{\gamma}_3, \bar{\eta})^T \in \left\{ k \cdot \left( 1, \frac{7}{27}, \frac{2}{9}, \frac{70}{189} \right)^t : k \in [9/2 \cdot \epsilon_0, 1] \right\}$$

einen Gleichgewichtszustand darstellt und dass die eben aufgeschriebene Menge alle möglichen Fixpunkte enthält.

### Stabilität BC

Wir stellen wiederum nach längerer Rechnung fest, dass der Vektor

$$\bar{\gamma} = \epsilon_0 \cdot (27/8, 21/20, 1)^T$$

ein Gleichgewichtszustand ist. Um das nachzuprüfen, bestimmen wir die dazugehörige mittlere Projekteffizienz

$$\bar{\eta} = \left( \sum_{r \in \bar{\mathcal{R}}} \frac{d_r}{\bar{\gamma}_r} \right)^{-1} = \epsilon_0 \cdot \left( \frac{0.45}{27/8} + \frac{0.35}{21/20} + \frac{0.2}{1} \right)^{-1} = \epsilon_0 \cdot \left( \frac{40 + 100 + 60}{300} \right)^{-1} = \epsilon_0 \cdot \frac{3}{2}$$

sowie den Vektor  $\Delta \bar{z}$ :

$$\Delta \bar{z}_1 = -0.1 + 0.5 \cdot \frac{0.45}{27/8} \cdot 3/2 = 0 \quad \checkmark$$

$$\Delta \bar{z}_2 = -0.2 + 0.4 \cdot \frac{0.35}{21/20} \cdot 3/2 = 0 \quad \checkmark$$

$$\Delta \bar{z}_3 = -0.3 + 0.9 \cdot \frac{0.2}{1} \cdot 3/2 = -\frac{3}{100} < 0 \quad \checkmark$$

Nach Eindeutigkeitsatz ABC ist der Vektor  $\bar{\gamma}$  der eindeutige Gleichgewichtszustand. Da aber  $\epsilon_0$  winzig klein ist, ist dieser Fixpunkt dem (Fast)stillstand gleichzusetzen.

Wir fassen die Ergebnisse in Tabelle 7 zusammen und stellen sie in Abbildung 18 auf der nächsten Seite grafisch dar.

$r$	Komp.	Stabilität A			Stabilität AB			Stabilität B			Stabilität BC		
		$\beta_r$	$\alpha_r$	$\bar{\gamma}_r$	$\beta_r$	$\alpha_r$	$\bar{\gamma}_r$	$\beta_r$	$\alpha_r$	$\bar{\gamma}_r$	$\beta_r$	$\alpha_r$	$\bar{\gamma}_r$
1	Schw.	0.3	0.7	1.0	0.1	0.7	1.00	0.1	0.6	1.00	0.1	0.5	$27/8 \epsilon_0$
2	Radf.	0.1	0.4	1.0	0.2	0.4	0.26	0.2	0.4	0.26	0.2	0.4	$21/20 \epsilon_0$
3	Laufen	0.1	0.9	1.0	0.3	0.9	0.22	0.3	0.9	0.22	0.3	0.9	$\epsilon_0$

Tabelle 7: Gleichgewichtszustände im Stabilitätsbeispiel. Bei Stabilität B gibt es unendlich viele Gleichgewichtszustände der Form  $k \cdot \bar{\gamma}$ ,  $k \in [9/2 \cdot \epsilon_0, 1]$ . Und bei Stabilität BC ist Quasistillstand im Fixpunkt gegeben, da  $\epsilon_0$  winzig klein ist.

### 5.3. Berechnung des Gleichgewichtszustands

#### 5.3.1. Das Gleichungssystem

Gegeben ist eine zulässige Mischung  $\bar{x} \in \bar{\mathcal{S}}$ . Wir definieren also  $\bar{\mathcal{N}}$  und  $\bar{\mathcal{R}}$  so, wie in (78) - (79) auf Seite 57 beschrieben.

Das Herzstück unserer Überlegungen ist Definition 27 auf Seite 59, welche uns die Stabilitätsbedingungen für die Kompetenzen liefert. Basierend auf der aktuellen Lösung für  $\bar{\gamma}$  bestimmen wir die Ausdrücke

$$\Delta \bar{z}_r = -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \bar{x}_i \quad (103)$$

und teilen die Kompetenzen wie in Abschnitt 5.2.5 in folgende drei Mengen auf:

$$\bar{\mathcal{R}}_A = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0\} \quad (104)$$

$$\bar{\mathcal{R}}_B = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r \in [\epsilon_0, 1] \wedge \Delta \bar{z}_r = 0\} \quad (105)$$

$$\bar{\mathcal{R}}_C = \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0\} \quad (106)$$

Dabei ist  $\epsilon_0 > 0$  eine winzig kleine Zahl (zB  $10^{-10}$ ), die sicherstellt, dass die Ausdrücke

$$\bar{d}_{ir} = \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i$$

stets berechenbar bleiben. Wir haben bereits angesprochen, dass dies keinen nennenswerten Fehler produziert und stellen fest, dass der Rechenfehler beliebig klein gemacht werden kann, wenn man nur  $\epsilon_0$  entsprechend klein wählt.

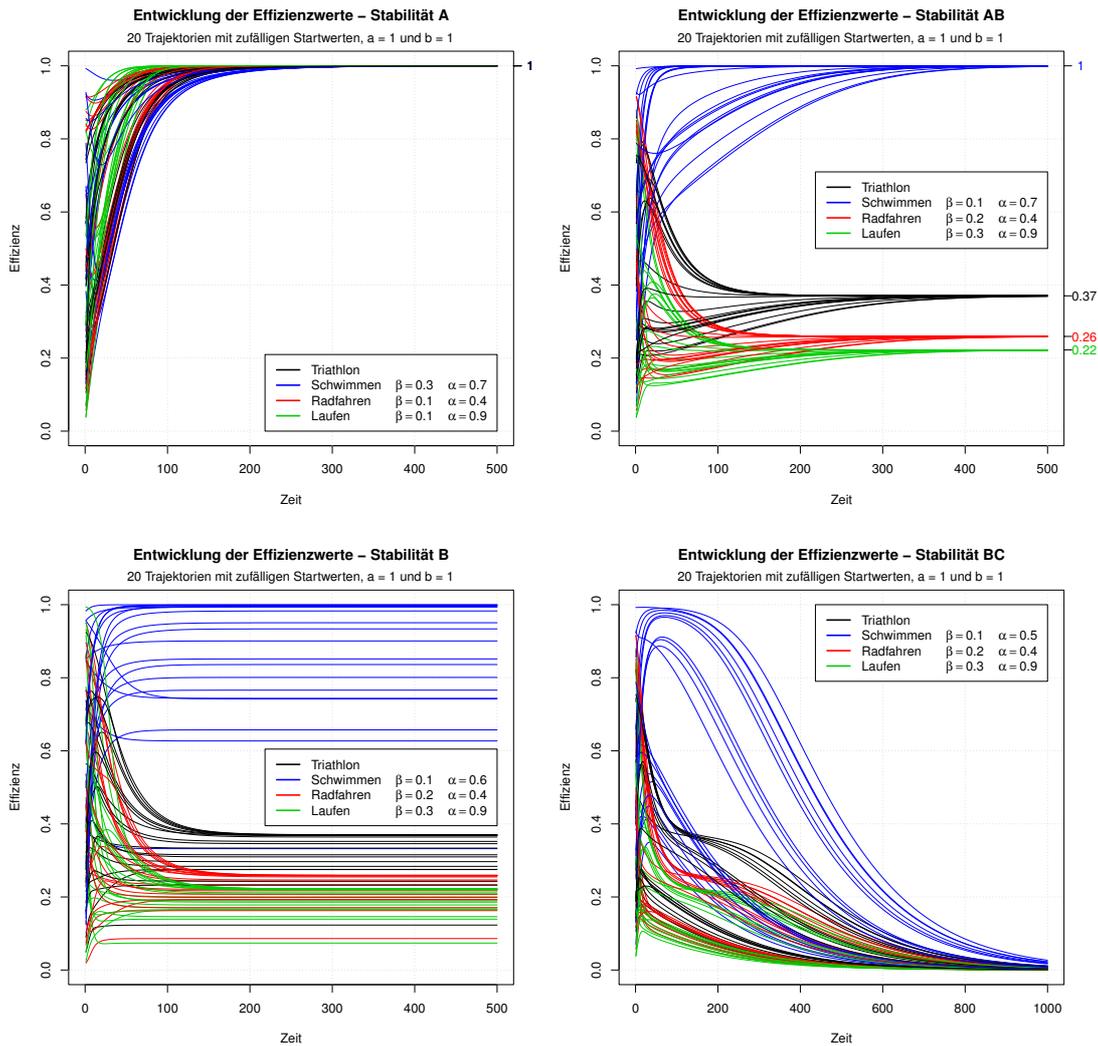


Abbildung 18: Entwicklung der Effizienzwerte im Stabilitätsbeispiel.

**Links oben:** Völlige Perfektion ist der eindeutige Fixpunkt. Das Kompetenzentwicklungssystem strebt gegen ihn, egal welche Startwerte wir vorgeben. **Rechts oben:** Für Schwimmen gilt Stabilität A, für die beiden anderen Disziplinen Stabilität B. Wieder entwickeln sich die Effizienzwerte stets in Richtung des eindeutigen Fixpunktes. **Links unten:** Da Stabilität B für alle Kompetenzen gilt, sind lediglich die relativen Beziehungen zwischen den  $\bar{\gamma}_r$  eindeutig, was das Gleichgewicht angeht. Wir stellen fest, dass die Effizienzwerte allesamt (abhängig von den Startwerten) gegen einen Fixpunkt streben. **Rechts unten:** Da für Laufen Stabilität C gilt und  $\epsilon_0$  beliebig klein gemacht werden kann, ist der (Quasi)stillstand der einzige Fixpunkt, dem sich alle Trajektorien zubewegen.

Für ein gegebenes  $\bar{\gamma}$  ist also das folgende **Gleichungssystem** zu lösen:

$$\bar{\gamma}_r = 1 \quad \forall r \in \bar{\mathcal{R}}_A \quad (107)$$

$$\bar{\gamma}_r = \epsilon_0 \quad \forall r \in \bar{\mathcal{R}}_C \quad (108)$$

$$0 = \Delta \bar{z}_r = -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \bar{d}_{ir} \bar{x}_i = -\beta_r + \alpha_r \sum_{i \in \bar{\mathcal{N}}} \frac{\bar{d}_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \bar{x}_i \quad \forall r \in \bar{\mathcal{R}}_B \quad (109)$$

Ziel ist es, eine hinreichend gute Lösung für dieses Gleichungssystem zu finden, bei der gleichzeitig die Kompetenzen korrekt den Mengen  $\bar{\mathcal{R}}_A$ ,  $\bar{\mathcal{R}}_B$  und  $\bar{\mathcal{R}}_C$  zugeordnet sind, das heißt, es zu keinerlei Verschiebungen kommt.

Wegen der auftretenden Summe und der Gestalt von  $\bar{\eta}_i$  haben wir es in der Regel mit einem nichtlinearen Gleichungssystem in  $\bar{\gamma}$  zu tun, das wir leider nicht analytisch lösen können. Wir wünschen uns ein Verfahren, das

- schneller arbeitet, als reines Durchiterieren, vorausgesetzt, das Kompetenzentwicklungssystem konvergiert gegen einen Fixpunkt,
- vergleichbare Resultate zwischen verschiedenen Grenzportfolios ermöglicht, falls Stabilität B für alle Kompetenzen eines Inselbetriebs gilt und
- sicher mit einem Gleichgewichtszustand (der immer existiert) terminiert.

Im folgenden Abschnitt besprechen wir eine relativ einfache Methode, welche unseren Wünschen weitgehend gerecht wird.

### 5.3.2. Die „Holzhackermethode“

#### *Grundidee*

Eines gleich vorweg: Den Namen habe ich frei erfunden. Der Name rührt daher, dass man sich die Effizienzwerte  $\bar{\gamma}_r$  als Baumstämme vorstellen kann, dessen Längen zunächst jeweils 1 betragen. Ziel ist es nun, von jedem Stamm gerade so viel Holz abzuhacken, sodass jede Kompetenz der richtigen Menge  $\bar{\mathcal{R}}_A$ ,  $\bar{\mathcal{R}}_B$  oder  $\bar{\mathcal{R}}_C$  zugeordnet ist und das Gleichungssystem (107)-(109) bezüglich des Genauigkeitsparameters  $\epsilon > 0$  hinreichend genau gelöst ist.

Zu Beginn setzen wir  $\bar{\gamma}_r = 1 \quad \forall r \in \bar{\mathcal{R}}$ , das heißt, alle Kompetenzen liegen in  $\bar{\mathcal{R}}_A$ . Wir konstruieren eine monoton fallende Folge  $\bar{\gamma}^{(k)}$ , die von oben gegen die Lösung konvergiert, beginnend mit  $k = 1$ .

Wir sägen für die Konstruktion von  $\bar{\gamma}^{(k)}$  solange an den geeigneten Stellen Holz der Länge  $1/2^k$  weg, bis wir entweder eine hinreichend genaue Lösung gefunden haben oder feststellen, dass wir zu grob geschnitten haben. Trifft Zweites zu, so setzen wir

$$k \leftarrow k + 1$$

dh wir halbieren die Schnittlänge. Wie ein Bildhauer, der seinen Stein zunächst grob in eine Form bringt und dann immer feinere Werkzeuge verwendet, um dem Stein den nötigen Feinschliff zu geben.

### Der Ablauf

In jeder Iteration berechnen wir zunächst  $\Delta \bar{z}_r \forall r \in \bar{\mathcal{R}}$ . Wir können die Prozedur mit einer hinreichend guten Lösung  $\bar{\gamma}$  **abbrechen**, falls folgende Bedingungen erfüllt sind:

$$\Delta \bar{z}_r > 0 \quad \forall r \in \bar{\mathcal{R}}_A \quad (110)$$

$$|\Delta \bar{z}_r| < \epsilon \quad \forall r \in \bar{\mathcal{R}}_B \quad (111)$$

$$\Delta \bar{z}_r < 0 \quad \forall r \in \bar{\mathcal{R}}_C \quad (112)$$

Wir können nie sagen, wann wir zu viel Holz abgehackt haben. Wenn aber der Umstand

$$\Delta \bar{z}_r < \epsilon \quad \forall r \in \bar{\mathcal{R}}_B \quad (113)$$

zutrifft (trifft immer zu, falls  $\bar{\mathcal{R}}_B$  leer ist), so haben wir definitiv nicht zu viel Holz erwischt. Denn in diesem Fall kann eine abbruchreife Lösung alleine durch Senken der  $\bar{\gamma}_r$ -Werte erwirkt werden, die Erhöhung eines solchen ist nicht notwendig.

Daher können wir jetzt (und nur jetzt) die **Mengen**  $\bar{\mathcal{R}}_A$ ,  $\bar{\mathcal{R}}_B$  und  $\bar{\mathcal{R}}_C$  gemäß (104)-(106) **aktualisieren**. Weiters **sichern** wir diesen Zustand, um ihn bei Bedarf laden zu können:

$$\bar{\gamma}^{(k)} \leftarrow \bar{\gamma} \quad (114)$$

Falls nun die Bedingung

$$\exists r \in \bar{\mathcal{R}}_B : \Delta \bar{z}_r \geq \epsilon \wedge \{\rho \in \bar{\mathcal{R}}_B : \bar{\gamma}_\rho > \epsilon_0\} \cap \{\rho \in \bar{\mathcal{R}}_B : \Delta \bar{z}_\rho \leq -\epsilon\} = \emptyset \quad (115)$$

zutrifft, so haben wir höchstwahrscheinlich zu grob gesäbelt. Die Menge

$$\{\rho \in \bar{\mathcal{R}}_B : \bar{\gamma}_\rho > \epsilon_0\} \quad (116)$$

enthält all jene Kompetenzen, dessen Effizienzwerte im Augenblick senkbar sind. Denn Kompetenzen der Menge  $\bar{\mathcal{R}}_A$  dürfen solange nicht angetastet werden, bis ein Update und eine damit verbundene Verschiebung in die Menge  $\bar{\mathcal{R}}_B$  erfolgt ist. Die Menge

$$\{\rho \in \bar{\mathcal{R}}_B : \Delta \bar{z}_\rho \leq -\epsilon\} \quad (117)$$

beinhaltet all jene Kompetenzen der Menge  $\bar{\mathcal{R}}_B$ , die eine Evidenz für eine Senkung aufweisen. Ist der Schnitt beider Mengen leer, so gibt es momentan keine Effizienzwerte, die wir senken dürfen, wegen  $\Delta \bar{z}_r \geq \epsilon$  für ein  $r \in \bar{\mathcal{R}}_B$  sehr wohl aber ein  $\bar{\gamma}_r$ , das wir erhöhen müssten.

Folglich **laden** wir den gesicherten Zustand und **verfeinern die Suche**:

$$\bar{\gamma} \leftarrow \bar{\gamma}^{(k)} \quad (118)$$

$$k \leftarrow k + 1 \quad (119)$$

Falls Bedingung (115) nicht zutrifft, können wir unter Vorbehalt fröhlich weitersäbeln:

$$\bar{\gamma}_r \leftarrow \max(\bar{\gamma}_r - 1/2^k, \epsilon_0) \quad \forall r \in \bar{\mathcal{R}}_B : \Delta \bar{z}_r \leq -\epsilon \quad (120)$$

---

**Algorithmus 6** Berechne Gleichgewichtszustand – Holzhackermethode

---

**Input:**  $\bar{x}$  – das gewünschte Grenzportfolio,  $\epsilon > 0$  – der Genauigkeitsparameter,  $\epsilon_0 > 0$   
– der Mindestwert für  $\bar{\gamma}_r$

**Output:**  $L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  – die Lösungsliste zu  $\bar{x}$

```
1:  $\bar{\gamma}_r \leftarrow 1 \quad \forall r \in \bar{\mathcal{R}}$ 
2:  $\bar{\gamma}_r \leftarrow \epsilon_0 \quad \forall r \in \mathcal{R}/\bar{\mathcal{R}}$ 
3:  $k \leftarrow 1$  // Die Genauigkeitsstufe
4:  $\bar{\mathcal{R}}_A \leftarrow \bar{\mathcal{R}}$ 
5:  $\bar{\mathcal{R}}_B \leftarrow \emptyset$ 
6:  $\bar{\mathcal{R}}_C \leftarrow \emptyset$ 
7:  $\bar{\gamma}^S \leftarrow \bar{\gamma}$  // Der letztmalig gesicherte Zustand
8: loop
9:    $\bar{\eta}_i \leftarrow \left( \sum_{r \in \mathcal{R}_i} \frac{d_{ir}}{\bar{\gamma}_r} \right)^{-1} \quad \forall i \in \mathcal{N}$ 
10:   $\Delta \bar{z}_r \leftarrow -\beta_r + \alpha_r \sum_{i \in \mathcal{N}} \frac{d_{ir}}{\bar{\gamma}_r} \bar{\eta}_i \bar{x}_i \quad \forall r \in \bar{\mathcal{R}}$ 
11:  if  $\Delta \bar{z}_r > 0 \quad \forall r \in \bar{\mathcal{R}}_A \wedge |\Delta \bar{z}_r| < \epsilon \quad \forall r \in \bar{\mathcal{R}}_B \wedge \Delta \bar{z}_r < 0 \quad \forall r \in \bar{\mathcal{R}}_C$  then
12:     $\bar{w} \leftarrow \sum_{i \in \mathcal{N}} w_i \bar{\eta}_i \bar{x}_i$  // Abbruch
13:    return  $L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$ 
14:  end if
15:  if  $\Delta \bar{z}_r < \epsilon \quad \forall r \in \bar{\mathcal{R}}_B$  then
16:     $\bar{\gamma}^S \leftarrow \bar{\gamma}$  // Zustand sichern
17:     $\bar{\mathcal{R}}_A \leftarrow \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = 1 \wedge \Delta \bar{z}_r > 0\}$  // Mengenupdate
18:     $\bar{\mathcal{R}}_C \leftarrow \{r \in \bar{\mathcal{R}} : \bar{\gamma}_r = \epsilon_0 \wedge \Delta \bar{z}_r < 0\}$ 
19:     $\bar{\mathcal{R}}_B \leftarrow \bar{\mathcal{R}} / (\bar{\mathcal{R}}_A \cup \bar{\mathcal{R}}_C)$ 
20:  end if
21:  if  $\exists r \in \bar{\mathcal{R}}_B : \Delta \bar{z}_r \geq \epsilon \wedge \{\rho \in \bar{\mathcal{R}}_B : \bar{\gamma}_\rho > \epsilon_0\} \cap \{\rho \in \bar{\mathcal{R}}_B : \Delta \bar{z}_\rho \leq -\epsilon\} = \emptyset$  then
22:     $\bar{\gamma} \leftarrow \bar{\gamma}^S$  // Zustand laden
23:     $k \leftarrow k + 1$  // Suche verfeinern
24:  else
25:     $\bar{\gamma}_r \leftarrow \max(\bar{\gamma}_r - 1/2^k, \epsilon_0) \quad \forall r \in \bar{\mathcal{R}}_B : \Delta \bar{z}_r \leq -\epsilon$  // Senken
26:  end if
27: end loop
```

---

### Der Algorithmus

Diese Idee ist in Algorithmus 6 auf der vorherigen Seite aufgeschrieben. Der Gleichgewichtszustand hängt also nicht von den Parametern  $a$  und  $b$  der Lernfunktion  $\bar{\varphi}(\cdot)$  ab.

**Satz 44:** *Algorithmus 6 terminiert in endlicher Laufzeit.*

*Beweis.* Wir haben oben argumentiert, dass wir von jedem gesicherten Zustand  $\bar{\gamma}^S$  aus stets eine abbruchreife Lösung erhalten können, ohne ein  $\bar{\gamma}_r$  erhöhen zu müssen. Sei  $\bar{\gamma}^S$  ein beliebiger gesicherter Zustand. Wir stellen fest, dass es  $k^* \in \mathbb{N}$  und Zahlen  $\lambda_r \in \mathbb{N}_0$  gibt, sodass der Vektor

$$\bar{\gamma} = \left( \bar{\gamma}_1^S - \frac{\lambda_1}{2^{k^*}}, \bar{\gamma}_2^S - \frac{\lambda_2}{2^{k^*}}, \dots, \bar{\gamma}_R^S - \frac{\lambda_R}{2^{k^*}} \right)^T$$

eine abbruchreife Lösung darstellt. Die Zahlen  $\lambda_r$  entsprechen der Anzahl der notwendigen Subtraktionen bei der Schrittlänge  $1/2^{k^*}$ . Dabei muss  $k^*$  nur hinreichend groß sein.

Gilt in Algorithmus 6 nach Zeile 10  $\Delta \bar{z}_r < \epsilon \forall r \in \bar{\mathcal{R}}_B$ , so brechen wir entweder ab oder sichern. Der Nichtabbruch wird durch mindestens einen der folgenden Umstände erwirkt:

1.  $\exists r \in \bar{\mathcal{R}}_B : \Delta \bar{z}_r \leq -\epsilon$
2.  $\exists r \in \bar{\mathcal{R}}_A : \Delta \bar{z}_r \leq 0$

Im 1. Fall wird Kompetenz  $r$  entweder von  $\bar{\mathcal{R}}_B$  nach  $\bar{\mathcal{R}}_C$  verschoben, falls  $\bar{\gamma}_r = \epsilon_0$ , oder  $\bar{\gamma}_r$  wird gesenkt, falls  $\bar{\gamma}_r > \epsilon_0$ . Im 2. Fall wird Kompetenz  $r$  von  $\bar{\mathcal{R}}_A$  nach  $\bar{\mathcal{R}}_B$  verschoben.

Gibt es nach Zeile 10 ein  $r \in \bar{\mathcal{R}}_B$ , sodass  $\Delta \bar{z}_r \geq \epsilon$ , dann wird entweder  $k$  erhöht, falls  $\{\rho \in \bar{\mathcal{R}}_B : \bar{\gamma}_\rho > \epsilon_0\} \cap \{\rho \in \bar{\mathcal{R}}_B : \Delta \bar{z}_\rho \leq -\epsilon\} = \emptyset$  oder mindestens ein  $\bar{\gamma}_r$  gesenkt.

Der Algorithmus kann also nicht stecken bleiben. Ein Mengenupdate ist höchstens  $2 \cdot R$  Mal möglich. Das Senken von Effizienzwerten ist für jedes  $k$  nur endlich oft möglich. Solange also  $k$  zu niedrig ist, wird früher oder später im Zuge des Ladens  $k$  erhöht. Über kurz oder lang sind die Subtraktionsvorgänge also fein genug.  $\square$

**Beispiel 45:** *Wir veranschaulichen Algorithmus 6 an Hand des folgenden einfachen Beispiels mit  $n = 2$  Projektklassen und  $R = 3$  Kompetenzen:*

$$d = \begin{bmatrix} 0 & 5/6 & 1/6 \\ 3/5 & 0 & 2/5 \end{bmatrix} \quad \beta = \begin{pmatrix} 3/8 \\ 1/5 \\ 1/8 \end{pmatrix} \quad \alpha = \begin{pmatrix} 5/8 \\ 9/10 \\ 3/4 \end{pmatrix} \quad \bar{x} = \begin{pmatrix} 1/4 \\ 3/4 \end{pmatrix} \quad \sigma = 1$$

*Auf eine genaue Rechnung verzichten wir an dieser Stelle, stattdessen verweisen wir auf Abbildung 19 auf Seite 88, in welcher alles genau erklärt wird.*

*Wir beschränken uns hier lediglich auf die Kontrolle, ob der Endzustand*

$$\bar{\gamma} = (3/8, 5/8, 1)^T \quad \text{sowie} \quad \bar{\eta} = (2/3, 1/2)^T$$

*tatsächlich das System (107)-(109) löst.*

$$\Delta \bar{z}_1 = -\beta_1 + \alpha_1 \sum_{i=1}^2 \frac{d_{i1}}{\bar{\gamma}_1} \bar{\eta}_i \bar{x}_i = -\frac{3}{8} + \frac{5}{8} \cdot \frac{3/5}{3/8} \cdot \frac{1}{2} \cdot \frac{3}{4} = 0 \quad \checkmark$$

$$\Delta \bar{z}_2 = -\beta_2 + \alpha_2 \sum_{i=1}^2 \frac{d_{i2}}{\bar{\gamma}_2} \bar{\eta}_i \bar{x}_i = -\frac{1}{5} + \frac{9}{10} \cdot \frac{5/6}{5/8} \cdot \frac{2}{3} \cdot \frac{1}{4} = 0 \quad \checkmark$$

$$\Delta \bar{z}_3 = -\beta_3 + \alpha_3 \sum_{i=1}^2 \frac{d_{i3}}{\bar{\gamma}_3} \bar{\eta}_i \bar{x}_i = -\frac{1}{8} + \frac{3}{4} \cdot \frac{1/6}{1} \cdot \frac{2}{3} \cdot \frac{1}{4} + \frac{3}{4} \cdot \frac{2/5}{1} \cdot \frac{1}{2} \cdot \frac{3}{4} = \frac{25}{3000} > 0 \quad \checkmark$$

Voilà!

## 5.4. Bestimmung lokaler Grenzoptima

### 5.4.1. Motivation

Die optimale Entscheidung im Triathlonbeispiel sieht so aus: Wir sollten unsere gesamte Zeit mit Triathlon verbringen.

Einige werden jetzt denken, dass für diese Entscheidung keine komplizierte Rechnung notwendig sei, ist doch Triathlon die einzige Projektklasse. Um für ein wenig mehr Spannung zu sorgen, erweitern wir nun das Triathlon-Beispiel um zwei weitere Projektklassen (Sportarten):

i = 1: Triathlon (Schwimmen + Radfahren + Laufen) – wie gehabt

i = 2: Duathlon (Radfahren + Laufen)

i = 3: Laufen

Zusätzlich führen wir Erträge  $w_i$  für jede Projektklasse ein, welche wir zB als Spaßfaktor interpretieren können. Je besser wir in einer Sportart sind und je mehr Disziplinen darin vorkommen, desto mehr Spaß macht uns dieser Sport.

Die folgenden Parameterfestlegungen bezeichnen wir als **übliche Parameterkonstellation**. Wann immer nichts anderes angegeben ist, kommen sie zur Geltung.

$$d = \begin{bmatrix} 0.45 & 0.35 & 0.2 \\ 0 & 0.8 & 0.2 \\ 0 & 0 & 1 \end{bmatrix} \quad \beta = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix} \quad \alpha = \begin{pmatrix} 0.7 \\ 0.4 \\ 0.9 \end{pmatrix} \quad w = \begin{pmatrix} 300 \\ 200 \\ 100 \end{pmatrix} \quad \sigma = 1$$

Wir berechnen für alle drei Projektklassen die Effizienzwerte im Gleichgewicht, vorausgesetzt, wir stecken unsere gesamte verfügbare Zeit in die jeweilige Sportart:

$$\bar{\eta}_1 = \frac{70}{189} \approx 0.37 \quad \bar{\eta}_2 = \frac{5}{6} \approx 0.83 \quad \bar{\eta}_3 = 1$$

sowie die entsprechenden Grenzgewinne

$$\bar{w}_1 = w_1 \cdot \bar{\eta}_1 \approx 111.11 \quad \bar{w}_2 = w_2 \cdot \bar{\eta}_2 \approx 166.67 \quad \bar{w}_3 = w_3 \cdot \bar{\eta}_3 = 100$$

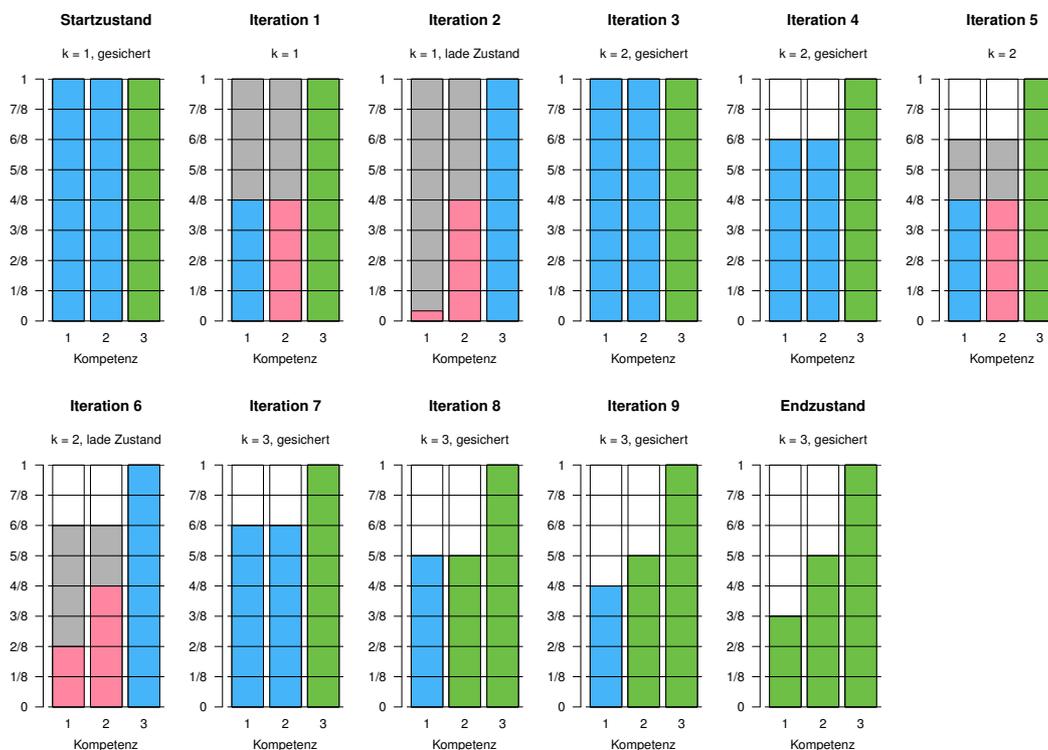


Abbildung 19: Veranschaulichung der Holzhackermethode in Beispiel 45. Grüne Balken bedeuten, dass sich die entsprechenden Effizienzwerte  $\bar{\gamma}_r$  im abbruchreifen Zustand befinden, während blaue bzw. rote Balken darauf hinweisen, dass die entsprechenden Effizienzwerte gesenkt bzw. erhöht werden sollten. Der letztmalig gesicherte Zustand ist grau markiert. In Iteration 1 (wie auch in Iteration 5) gilt zwar  $\Delta\bar{z}_2 \geq \epsilon$ , aber die Effizienz der ersten Kompetenz kann noch gesenkt werden. In Iteration 2 erreicht  $\bar{\gamma}_1$  den Wert  $\epsilon_0$  und da die Bedingung in Zeile 21 von Algorithmus 6 erfüllt ist, muss erstmals  $k$  erhöht und der gesicherte Zustand geladen werden. In Iteration 4 können wir den aktuellen Zustand sichern, da sowohl  $\Delta\bar{z}_1 < \epsilon$  als auch  $\Delta\bar{z}_2 < \epsilon$  gilt. In Iteration 6 wird aus dem gleichen Grund wie in Iteration 2 der gesicherte Zustand geladen und  $k$  erhöht. In Iteration 10 haben wir einen abbruchreifen Zustand aufgespürt.

Diese Rechnung würde uns nahe legen, sich auf den Duathlon zu konzentrieren. Allerdings stellt sich zumeist heraus, dass es Mischungen gibt, die einen zum Teil deutlich höheren Grenzgewinn ermöglichen. Abbildung 20 stellt die Grenzgewinne grafisch dar.

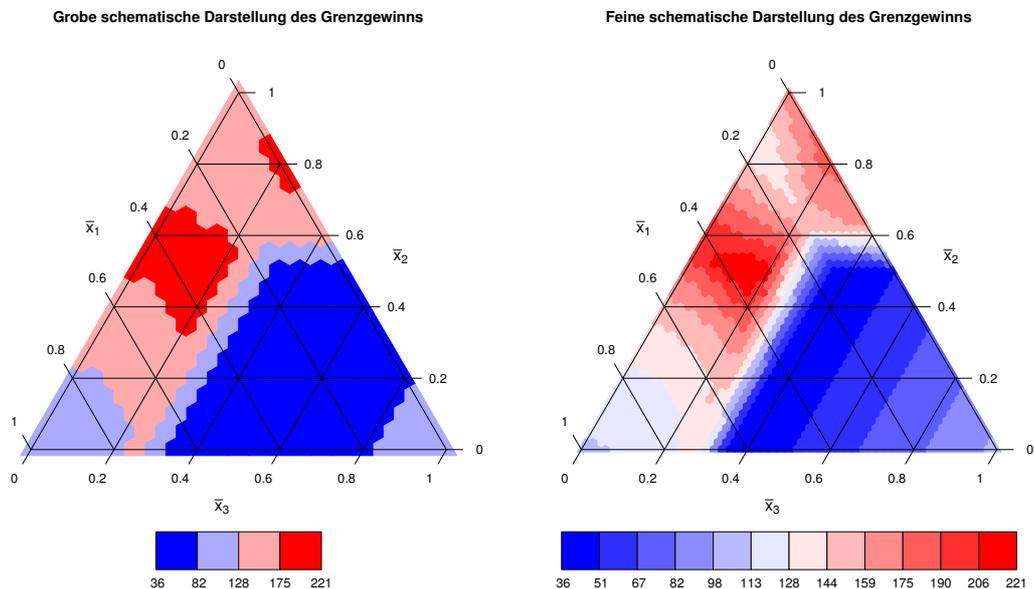


Abbildung 20: Schematische Darstellung des Grenzgewinns im Triathlonbeispiel.

**Links:** Wir erkennen zwei sehr gute Regionen, welche rot untermalt sind. Bei jener rechts oben handelt es sich um Mischungen zwischen Duathlon und Laufen, bei der anderen um Mischungen aller drei Projektklassen.

**Rechts:** Besonders auffällig ist das extrem starke Gefälle bei  $\bar{x}_3 \approx 0.3$ , was typisch für unser Modell ist. Offenbar fällt in dieser Region die Schwimm- bzw. Radfahrleistung derart stark ab, dass die Investitionen in Triathlon und Duathlon praktisch wertlos sind. Im Fall  $\bar{x}_1 \approx 0$  verschiebt sich dieses Gefälle nach rechts unten, da in solchen Portfolios Schwimmen gar nicht gebraucht wird.

Neben den Randpolitiken lohnt es sich also auch, Mischungen von Projektklassen zu betrachten. Unser Ziel ist es nun, den Grenzgewinn, welchen wir in Definition 42 auf Seite 77 eingeführt haben (lokal) zu maximieren:

$$\bar{w}(\bar{x}) = \sum_{i=1}^n w_i \bar{\eta}_i \bar{x}_i \rightarrow \max! \quad (121)$$

**Definition 46:** Ein Grenzportfolio  $\bar{x}^* \in \bar{S}$  heißt **lokales Grenzoptimum**, falls für jedes  $\bar{x} \in \bar{S}$ , welches hinreichend nahe  $\bar{x}^*$  ist, folgende Bedingung erfüllt ist:

$$\bar{w}(\bar{x}^*) > \bar{w}(\bar{x}) \quad (122)$$

**Beispiel 47:** Wir wissen, dass wir im Triathlonbeispiel mit der üblichen Parameterkonstellation von Seite 87 bessere Grenzgewinne erreichen können, wenn wir die Sportarten Triathlon, Duathlon und Laufen mischen.

In Abbildung 21 sind einige ausgewählte Punkte eingezeichnet und in Tabelle 8 auf der nächsten Seite charakterisieren wir die eingezeichneten Punkte.

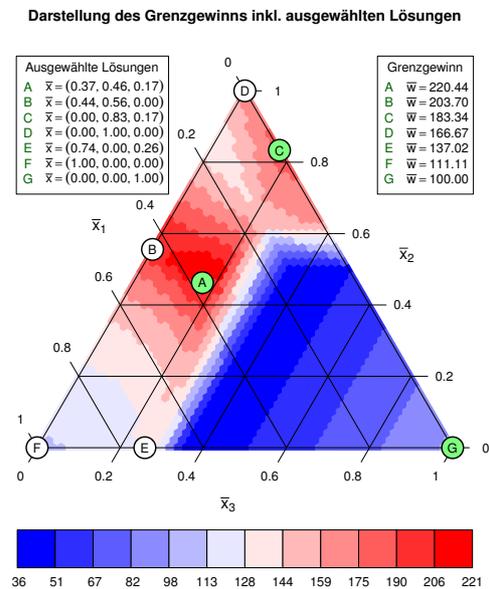


Abbildung 21: Grenzgewinne und ausgewählte Lösungen im Triathlonbeispiel in der üblichen Konstellation. Grün hinterlegte Punkte stellen die lokalen Grenzoptima dar. Die anderen Punkte bezeichnen die auf Teilmengen von  $\mathcal{N}$  eingeschränkten lokalen Grenzoptima. Punkt B ist zB die beste Mischung zwischen Triathlon und Duathlon, aber nicht lokal optimal. Punkt C ist die beste Mischung zwischen Duathlon und Laufen und gleichzeitig auch ein lokales Grenzoptimum. Punkt G ist die beste Mischung aus Laufen (no na) und gleichzeitig ein (schlechtes) lokales Grenzoptimum.

Wir erkennen zum Beispiel, dass die beste Mischung aus Triathlon und Duathlon (Punkt B) deutlich besser als das lokale Grenzoptimum G ist. Punkt C, das die beste Mischung aus Duathlon und Laufen markiert, ist ein lokales Grenzoptimum und gar nicht so übel.

Wir werden uns ein geeignetes ableitungsfreies Verfahren zunutzen machen, um lokale Grenzoptima aufzuspüren, das wir ansatzweise schon in Abschnitt 4.5 verwendet haben: Pattern Search.

Punkt	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{w}(\bar{x})$	$\bar{\mathcal{N}}$	$ \bar{\mathcal{N}} $	lokal optimal?
A	0.3710	0.4624	0.1666	220.40	{1, 2, 3}	3	Ja
B	0.4444	0.5556	0.0000	203.70	{1, 2}	2	Nein
C	0.0000	0.8334	0.1666	183.34	{2, 3}	2	Ja
D	0.0000	1.0000	0.0000	166.67	{2}	1	Nein
E	0.7407	0.0000	0.2593	137.02	{1, 3}	2	Nein
F	1.0000	0.0000	0.0000	111.11	{1}	1	Nein
G	0.0000	0.0000	1.0000	100.00	{3}	1	Ja

Tabelle 8: Übersicht über alle ausgewählten Punkte im Triathlonbeispiel

### 5.4.2. Pattern Search

#### *Grundidee*

Ausgehend vom aktuellen Grenzportfolio  $\bar{x}$  greifen wir uns zwei Projektklassen  $i, j \in \mathcal{N}$  heraus und untersuchen, wie sich der Grenzgewinn verändert, wenn wir für eine gegebene Schrittweite  $\lambda > 0$  folgende Modifikation vornehmen:

$$\bar{x}_i \leftarrow \bar{x}_i + \lambda \qquad \bar{x}_j \leftarrow \bar{x}_j - \lambda$$

Dies führen wir systematisch durch, um in der Umgebung

$$\mathcal{U}_\lambda(\bar{x}) = \{\bar{u} \in \mathbb{R}^n : \exists i, j \in \mathcal{N} : \bar{u}_i - \bar{x}_i = \lambda, \bar{u}_j - \bar{x}_j = -\lambda, \forall l \in \mathcal{N} / \{i, j\} : \bar{u}_l - \bar{x}_l = 0\}$$

nach dem besten „Nachbarn“ von  $\bar{x}$  bezüglich der Schrittweite  $\lambda$  zu suchen. Ist die so gefundene Lösung besser als die aktuelle, dann springen wir zu dieser, andernfalls verringern wir die Schrittlänge  $\lambda$ .

Wir fangen also mit einem großen Hyperstern an, mit dem wir den Raum zunächst grob durchsuchen und verfeinern dann sukzessive die Suche. Sobald  $\lambda$  den gegebenen Schwellenwert unterschreitet, brechen wir die Suche ab.

Abbildung 22 auf der nächsten Seite stellt diese Idee übersichtlich dar.

#### *Vermeidung von Doppelbesuchen*

In Abbildung 22 erkennen wir, dass einige Punkte in der Umgebung der aktuellen Lösung nicht blau, sondern grau dargestellt sind. Verwenden wir die selbe Schrittweite wie beim Vorgänger, so haben wir in unserer aktuellen Umgebung bereits einige Punkte vom Vorgänger aus besucht, welche wir uns nicht nochmals anschauen müssen (und wollen).

Es gibt  $\binom{n}{2}$  Möglichkeiten, zwei Projektklassen  $i, j \in \mathcal{N}$  auszuwählen. Dabei werden  $\bar{x}_i$  und  $\bar{x}_j$  jeweils ein Mal um  $\lambda$  erhöht bzw. gesenkt. Insgesamt gibt es also  $2 \cdot \binom{n}{2} = n \cdot (n - 1)$  Punkte in  $\mathcal{U}_\lambda(\bar{x})$ .

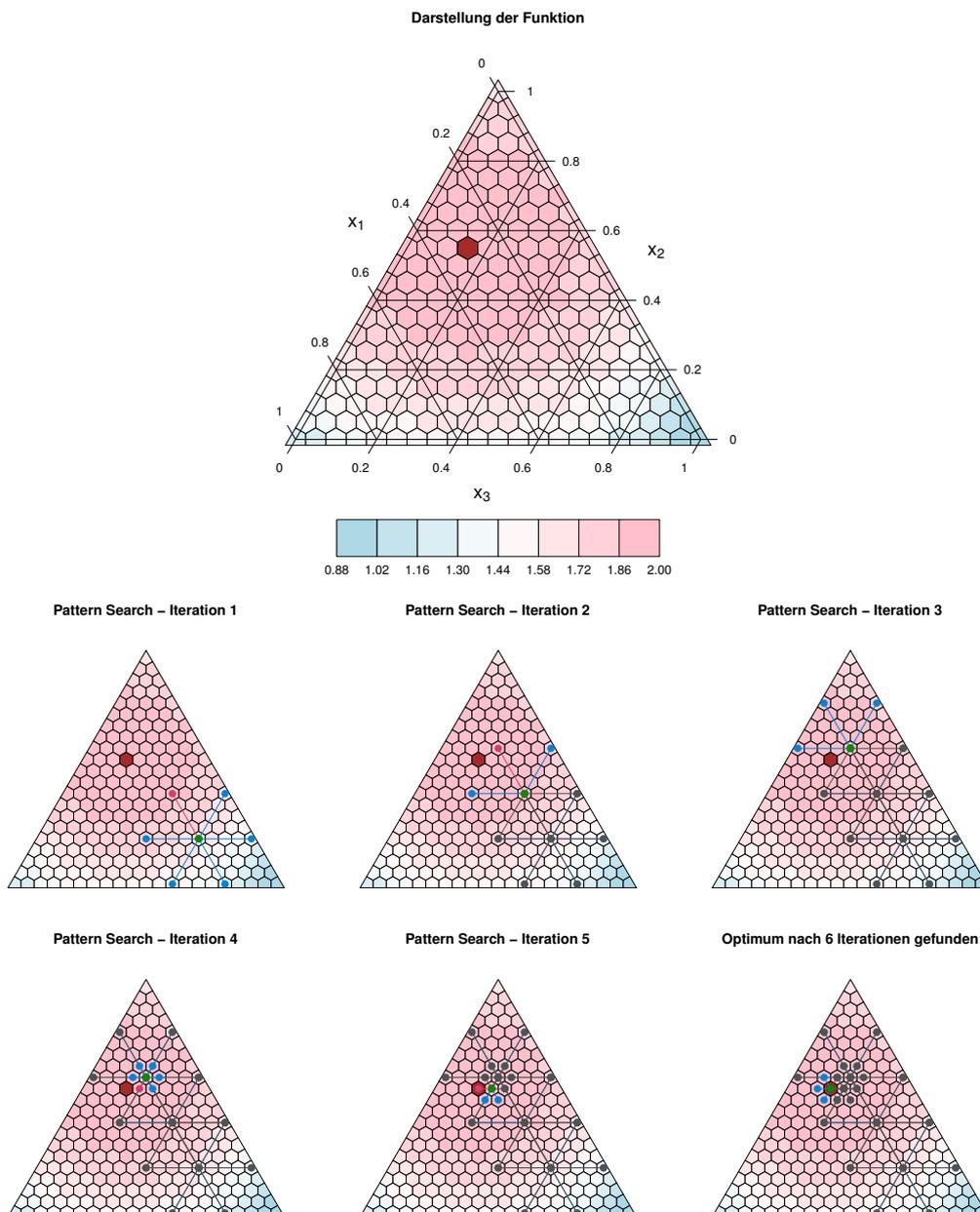


Abbildung 22: Veranschaulichung des Pattern Search Algorithmus. **Oben:** Darstellung der Funktion  $f(x) = 2 - (0.3 - x_1)^2 - (0.55 - x_2)^2 - (0.15 - x_3)^2$ . Das Optimum ist braun markiert. **Unten:** Wir beginnen unsere Suche bei  $\bar{x} = (0.2, 0.2, 0.6)^T$  mit  $\lambda = 0.2$ . Der aktuelle Punkt ist grün markiert. Wir besuchen alle Nachbarn (blau) und springen zum besten (rot). In Iteration 3 finden wir keine bessere Lösung in der Nachbarschaft, sodass wir  $\lambda = 0.05$  setzen und fortfahren, bis wir das Optimum gefunden haben.

Wurde beim Vorgänger  $\bar{x}_i$  erhöht und  $\bar{x}_j$  gesenkt, so kehren wir genau dann zu einem bereits besuchten Punkt zurück, wenn wir in der aktuellen Lösung entweder  $\bar{x}_i$  senken oder  $\bar{x}_j$  erhöhen (oder beides tun). Das sind  $n - 1 + n - 2 = 2n - 3$  Punkte, die wir uns also schenken können und es verbleiben lediglich  $n \cdot (n - 1) - (2n - 3) = n^2 - 3n + 3 = (n - 1) \cdot (n - 2) + 1$  zu durchsuchende Richtungen.

Sofern die Anzahl der Projektklassen nicht zu groß ist, ist die dadurch erzielte relative Ersparnis ziemlich groß, wie Tabelle 9 zeigt.

$ \mathcal{N} $	Anzahl Richtungen	Absolute Ersparnis	Relative Ersparnis	# bereinigte Richtungen
$n$	$n \cdot (n - 1)$	$2n - 3$	$\frac{2n-3}{n \cdot (n-1)}$	$n^2 - 3n + 3$
2	2	1	0.50	1
3	6	3	0.50	3
4	12	5	0.42	7
5	20	7	0.35	13
6	30	9	0.30	21
7	42	11	0.26	31
8	56	13	0.23	43
9	72	15	0.21	57
10	90	17	0.19	73

Tabelle 9: Einsparungspotenzial bei den Richtungen in Pattern Search

### Die Schrittlänge

Drei Fragen schreien nach einer Antwort:

- Wie hoch soll die Startschrittweite sein?
- Wie senken wir die Schrittweite?
- Wie hoch soll die minimale Schrittweite sein?

Einerseits soll die **Startschrittweite** groß gewählt werden, um schneller in gute Regionen vorzudringen. Wird diese aber zu groß gewählt, stoßen wir zu schnell an den Rand der zulässigen Menge. In diesem Fall könnte man quasi gleich von den Randlösungen aus starten, wodurch uns gute lokale Grenzoptima durch die Finger flutschen können. Ein guter Kompromiss für  $\sigma = 1$  ist der Startwert  $0.0256 = 4^4/1000$  – nicht zu groß und nicht zu klein.

Das **Senken der Schrittweite** erfolgt nach folgender Vorschrift:

$$\lambda \leftarrow \lambda \cdot \frac{1}{q}$$

Dabei ist  $q > 0$  der Faktor, durch den  $\lambda$  dividiert wird. Dies führen wir solange aus, bis die **minimale Schrittweite**  $\lambda_{min}$  unterschritten wird, also

$$\lambda < \lambda_{min}$$

gilt. In dieser Arbeit verwenden wir  $\lambda_{min} = 1/1000 = 0.001$ .

Oftmals wird  $q = 2$  gewählt, die Schrittweite also halbiert. Diverse Überlegungen und Versuche legen aber für unsere Zwecke nahe,  $q = 4$  zu wählen.

Begründung: Wir können von den eben angesprochenen Ersparnissen nur dann Gebrauch machen, wenn sich die Schrittweite nicht geändert hat, was bei  $q = 4$  deutlich öfter der Fall ist als bei  $q = 2$ .

### ***Randüberschreitungen***

Eine weitere wichtige Frage ist, wie wir darauf reagieren, wenn die Lösung beim Index  $j \in \mathcal{N}$  den Rand überschreitet, also  $\bar{x}_j - \lambda < 0$  ist. Es gibt zwei Varianten der Randüberschreitung, welche wir diskutieren müssen:

1. Für die aktuelle Lösung gilt  $\bar{x}_j > 0$ , sie befindet sich also im Inneren.
2. Für die aktuelle Lösung gilt  $\bar{x}_j = 0$ , sie liegt also bereits am Rand.

Im ersten Fall korrigieren wir die Lösung derart, dass der neue Punkt genau am Rand zum Erliegen kommt. Wir setzen exklusiv für diesen Punkt  $\lambda = \bar{x}_j$ . Falls dieser Punkt der beste Nachbar der aktuellen Lösung ist, müssen wir in der folgenden Iteration leider wieder alle Punkte in der Umgebung besuchen, da sich die Schrittweite geändert hat.

Im zweiten Fall freuen wir uns einfach darüber, dass wir diesen Punkt ignorieren können.

### ***Der Algorithmus***

In Algorithmus 7 auf der nächsten Seite schreiben wir den Pattern Search Algorithmus auf. Wie schon in den Algorithmen zur Berechnung des Gleichgewichtszustands fassen wir die Lösung zu einer Lösungsliste  $L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  zusammen. Auf einzelne Komponenten dieser Liste greifen wir mit dem Punkt-Operator zu.

### **5.4.3. Ebenen**

Algorithmus 7 liefert uns nicht unbedingt ein lokales Grenzoptimum, sondern nur einen (sehr) aussichtsreichen Kandidaten. Dazu betrachten wir drei Beispiele.

**Beispiel 48 (1. Ebenenbeispiel):** *Uns begegnet dieses simple Beispiel mit  $n = R = 2$ :*

$$d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \beta = \begin{pmatrix} 2/3 \\ 2/3 \end{pmatrix} \quad \alpha = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad w = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \sigma = 1$$

---

**Algorithmus 7** Berechne einen Kandidaten für lokales Grenzoptimum – Pattern Search

---

**Input:**  $L_S = L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  – die Startlösung,  $\lambda$  – die (Start)schrittweite,  $\lambda_{min}$  – Genauigkeitsschranke

**Output:**  $L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  – eine Lösungsliste mit einem Kandidaten für ein lokales Grenzoptimum ausgehend von  $L_S$

```
1:  $q \leftarrow 4$  // Schrumpfungsfaktor
2:  $L_A \leftarrow L_S$  // Aktuelle Lösung, zu Beginn Startlösung
3:  $L_N \leftarrow \text{Null}$  // Nachfolger - Next solution
4:  $i_P \leftarrow 0$ ;  $j_P \leftarrow 0$  // Woher kommt der Vorgänger? 0: kein Vorgänger
5:  $i_A \leftarrow 0$ ;  $j_A \leftarrow 0$  // Wohin geht die aktuelle Lösung? 0:  $\lambda$  gekürzt
6: loop
7:    $L_N \leftarrow L_A$  // Erhöhe  $\bar{x}_i$  und senke  $\bar{x}_j$ 
8:   for  $i \in \mathcal{N}$  do
9:     for  $j \in \{l \in \mathcal{N} / \{i\} : (L_A) \cdot \bar{x}_l > 0\}$  do
10:      if  $(L_A) \cdot \bar{x}_j - \lambda < 0$  then
11:         $\bar{x}_C \leftarrow (L_A) \cdot \bar{x}$ 
12:         $(\bar{x}_C)_i \leftarrow (L_A) \cdot \bar{x}_i + (L_A) \cdot \bar{x}_j$ ;  $(\bar{x}_C)_j \leftarrow 0$ 
13:         $L_C \leftarrow$  Berechne Gleichgewichtszustand gemäß Algorithmus 6
14:        if  $(L_C) \cdot \bar{w} > (L_N) \cdot \bar{w}$  then
15:           $L_N \leftarrow L_C$ ;  $i_A \leftarrow 0$ ;  $j_A \leftarrow 0$  // Bessere Lösung mit Randübertretung
16:        end if
17:      else
18:        if  $i \neq j_P \wedge j \neq i_P$  then
19:           $\bar{x}_C \leftarrow (L_A) \cdot \bar{x}$  // Noch nicht vom Vorgänger besucht
20:           $(\bar{x}_C)_i \leftarrow (L_A) \cdot \bar{x}_i + \lambda$ ;  $(\bar{x}_C)_j \leftarrow (L_A) \cdot \bar{x}_j - \lambda$ 
21:           $L_C \leftarrow$  Berechne Gleichgewichtszustand gemäß Algorithmus 6
22:          if  $(L_C) \cdot \bar{w} > (L_N) \cdot \bar{w}$  then
23:             $L_N \leftarrow L_C$ ;  $i_A \leftarrow i$ ;  $j_A \leftarrow j$  // Bessere Lösung ohne Randübertr.
24:          end if
25:        end if
26:      end if
27:    end for
28:  end for
29:  if  $L_N \neq L_A$  then
30:     $L_A \leftarrow L_N$ ;  $i_P \leftarrow i_A$ ;  $j_P \leftarrow j_A$  // Bessere Lösung in der Umgebung gefunden
31:    Continue
32:  end if
33:   $\lambda \leftarrow \lambda/q$  // Keine bessere Lösung in der Umgebung – Schrittweite senken
34:  if  $\lambda < \lambda_{min}$  then
35:    return  $L_A$ 
36:  end if
37:   $i_P \leftarrow 0$ ;  $j_P \leftarrow 0$  // Da  $\lambda$  gesenkt, gibt es offiziell keinen Vorgänger
38: end loop
```

---

Beide Projektklassen sind unabhängig. Nun folgt eine leichte Rechnung.

$$\Delta \bar{z}_1 = -\beta_1 + \alpha_1 \cdot \bar{d}_{11} \cdot \bar{x}_1 = -2/3 + 1 \cdot \bar{x}_1 > 0 \Leftrightarrow \bar{x}_1 > 2/3 \quad (123)$$

$$\Delta \bar{z}_2 = -\beta_2 + \alpha_2 \cdot \bar{d}_{22} \cdot \bar{x}_2 = -2/3 + 1 \cdot \bar{x}_2 > 0 \Leftrightarrow \bar{x}_2 > 2/3 \quad (124)$$

Offensichtlich gilt:  $\bar{\eta}_1 = \bar{\eta}_2 = \bar{w}(\bar{x}) = \epsilon_0$  für  $\bar{x}_1 \in (1/3, 2/3)$ , wie auch Abbildung 23 zeigt.

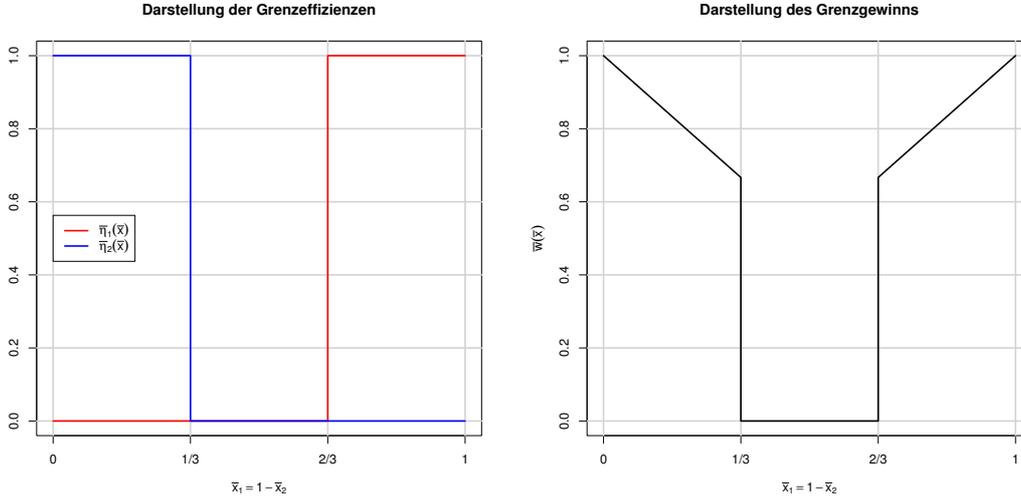


Abbildung 23: Grenzgewinne im 1. Ebenenbeispiel. **Links:** Darstellung der Grenzeffizienzen, die unsere Rechnung in (123)-(124) bestätigt: Nur wenn wir hinreichend viel Zeit in Projektklasse  $i$  investieren, können wir die entsprechende Grenzeffizienz  $\bar{\eta}_i$  auf 1 bringen. **Rechts:** Bei  $\bar{x}_1 = 1/3$  und  $\bar{x}_1 = 2/3$  gibt es eine Sprungstelle in  $\bar{w}(\bar{x})$ . Die optimale Entscheidung ist es, die gesamte Zeit entweder in Projektklasse 1 oder in Projektklasse 2 zu stecken.

Würden wir den Punkt  $\bar{x} = (0.5, 0.5)^T$  in Algorithmus 7 einsetzen, so würde er den Punkt (für  $\lambda < 1/6$ ) unverändert zurückgeben, da es keinen besseren Punkt in der Umgebung gibt. Leider ist diese vorgeschlagene Lösung ein absoluter Schwachsinn.

**Beispiel 49 (2. Ebenenbeispiel):** Ein zweites simples Beispiel mit  $n = R = 2$ :

$$d = \begin{bmatrix} 0 & 1 \\ 3/4 & 1/4 \end{bmatrix} \quad \beta = \begin{pmatrix} 1/5 \\ 1/5 \end{pmatrix} \quad \alpha = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \quad w = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad \sigma = 1$$

Abbildung 24 zeigt die Situation auf.

Wir können also in einer Ebene hängenbleiben, die wir sogleich formal definieren.

**Definition 50:** Eine **Ebene**  $\mathcal{E}$  ist ein zusammenhängender Bereich von zulässigen Grenzportfolios, sodass deren Grenzgewinne übereinstimmen, also folgendes gilt:

$$\bar{w}(\bar{x}) = \bar{w}(\bar{x}) \quad \forall \bar{x}, \bar{x} \in \mathcal{E} \quad (125)$$

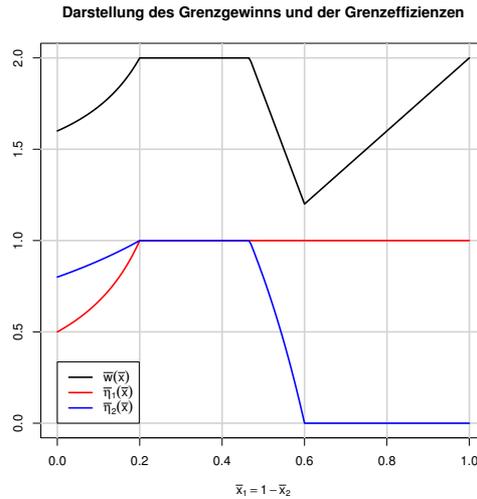


Abbildung 24: Grenzgewinne im 2. Ebenenbeispiel. Für  $\bar{x}_1 < 1/5$  gilt  $\bar{\gamma}_1 < 1$  und  $\bar{\gamma}_2 < 1$  und damit auch  $\bar{\eta}_1 < 1$  und  $\bar{\eta}_2 < 1$ . Für  $\bar{x}_1 > 7/15$  gilt  $\bar{\gamma}_1 < 1$  und somit auch  $\bar{\eta}_2 < 1$ . Falls  $\bar{x}_1 \in (1/5, 7/15)$  gilt, so können beide Projektklassen mit der maximalen Effizienz bearbeitet werden. Da  $w_1 = w_2$  gilt, ist jede Lösung im angegebenen Intervall gleich gut.

Wir unterscheiden dabei zwischen **Tiefebenen**, **Hochebenen** und **Zwischenebenen**. Im 1. Ebenenbeispiel haben wir es mit einer Tiefebene zutun, das heißt, wir verbessern uns jedenfalls, wenn wir die Ebene am Rand überspringen. Im 2. Ebenenbeispiel finden wir eine Hochebene vor: Überspringen wir den Rand dieser Ebene, so verschlechtern wir uns jedenfalls. Zwischenebenen sind ein Mittelding aus beidem. Ob Zwischenebenen überhaupt existieren können, bleibt in dieser Arbeit offen.

**Beispiel 51 (3. Ebenenbeispiel):** *Ein drittes Beispiel, diesmal etwas komplexer mit  $n = R = 3$ .*

$$\beta = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} \quad \alpha = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad w = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \sigma = 1$$

*Wir betrachten folgende zwei Varianten, die wir in Abbildung 25 veranschaulichen:*

$$d^{(1)} = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0 & 0.9 & 0.1 \\ 0.1 & 0 & 0.9 \end{bmatrix} \quad d^{(2)} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

Unser Pattern Search Algorithmus 7 liefert uns also entweder ein lokales Grenzoptimum, oder ein Grenzportfolio einer Ebene. Wir führen in Algorithmus 8 eine simple Möglichkeit an, wie wir überprüfen können, ob ein Grenzportfolio  $\bar{x}$  Teil einer Ebene ist.

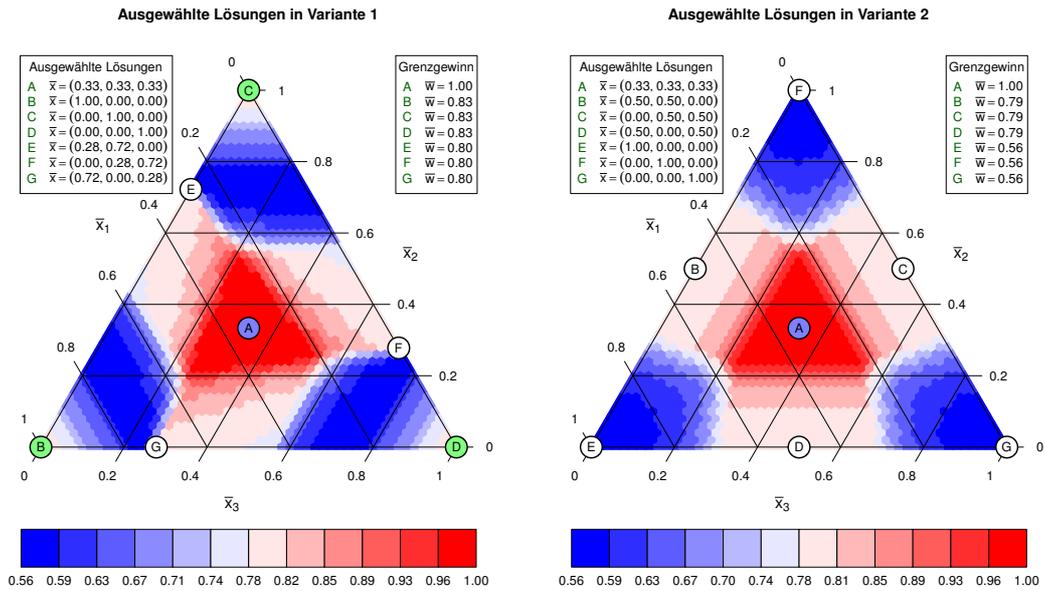


Abbildung 25: Grenzzugewinne und lokale Grenzoptima im 3. Ebenenbeispiel. **Blau** hinterlegte Punkte bezeichnen Punkte einer Ebene, **grün** hinterlegte Punkte lokale Grenzoptima und alle anderen Punkte sind wieder auf Teilmengen von  $\mathcal{N}$  eingeschränkte lokale Grenzoptima. Punkt A ist in beiden Varianten das Zentrum einer Hochebene. Während die Punkte B, C und D in Variante 1 lokale Grenzoptima darstellen, so gibt es technisch gesehen in Variante 2 keine lokalen Grenzoptima.

---

**Algorithmus 8** Prüfe auf Ebene

---

**Input:**  $L = L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  – zu überprüfende Lösung,  $\lambda_{min}$  – Genauigkeitsschranke

**Output:** gibt **True** zurück, falls  $\bar{x}$  bezüglich der Schrittweite  $\lambda_{min}$  Teil einer Ebene ist, andernfalls **False**

```
1: loop
2:   for  $i \in \mathcal{N}$  do
3:     for  $j \in \{l \in \mathcal{N} / \{i\} : L.\bar{x}_l > 0\}$  do
4:       if  $L.\bar{x}_j - \lambda_{min} < 0$  then
5:          $\bar{x}_C \leftarrow L.\bar{x}$ 
6:          $(\bar{x}_C)_i \leftarrow L.\bar{x}_i + L.\bar{x}_j$ 
7:          $(\bar{x}_C)_j \leftarrow 0$ 
8:          $L_C \leftarrow$  Berechne Gleichgewichtszustand gemäß Algorithmus 6
9:         if  $(L_C).\bar{w} = L.\bar{w}$  then
10:          return True // Teil einer Ebene
11:        end if
12:      else
13:         $\bar{x}_C \leftarrow L.\bar{x}$ 
14:         $(\bar{x}_C)_i \leftarrow L.\bar{x}_i + \lambda_{min}$ 
15:         $(\bar{x}_C)_j \leftarrow L.\bar{x}_j - \lambda_{min}$ 
16:         $L_C \leftarrow$  Berechne Gleichgewichtszustand gemäß Algorithmus 6
17:        if  $(L_C).\bar{w} = L.\bar{w}$  then
18:          return True // Teil einer Ebene
19:        end if
20:      end if
21:    end for
22:  end for
23:  return False // Kein Teil einer Ebene
24: end loop
```

---

## 5.5. Iterierte lokale Suche nach lokalen Grenzoptima

Wir erweitern unseren Pattern Search Algorithmus zu einem Iterated Local Search (ILS) und schreiben ihn in Algorithmus 9 auf. In diesem Algorithmus werden die Lösungen nicht auf Ebenen hin überprüft. Diesbezüglich verweisen wir auf Bemerkung 53 auf der nächsten Seite.

---

**Algorithmus 9** Generiere Kandidatenliste für lokale Grenzoptima – Pattern Search ILS

---

**Input:**  $\lambda$  – (Start)schrittweite,  $\lambda_{min}$  – Genauigkeitsschranke,  $N$  – Anzahl der Iterationen

**Output:**  $\mathcal{L}$  – Liste von Lösungslisten  $L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  mit lokalen Grenzoptima bzw. Lösungen von Ebenen

```

1: iter ← 1 // der Iterationszähler
2: repeat
3:    $\bar{x}$  ← Generiere gleichverteilte Zufallslösung auf  $\bar{S}$  gemäß Algorithmus 1
4:    $L$  ← Berechne Gleichgewichtszustand gemäß Algorithmus 6
5:    $L$  ← Berechne einen Kandidaten für lokales Grenzoptimum mit  $L^S = L$ 
      gemäß Algorithmus 7
6:   if  $L \notin \mathcal{L}$  then
7:     Füge  $L$  der Liste  $\mathcal{L}$  hinzu // Lösung noch nicht in der Liste
8:   end if
9:   iter ← iter + 1
10: until iter >  $N$ 
11: return  $\mathcal{L}$ 

```

---

Wir befassen uns abschließend noch kurz mit Attraktionsbecken.

**Definition 52:** Das **Attraktionsbecken** eines lokalen Grenzoptimums  $\bar{x}^*$  bzw. einer Ebene  $\mathcal{E}$  ist wie folgt definiert:

$$\mathcal{A}_\lambda(\bar{x}^*) := \{\bar{u} \in \bar{S} : \text{Alg. 7 retourniert } \bar{x}^* \text{ ausgehend von } \bar{u} \text{ mit Startschrittweite } \lambda\}$$

$$\mathcal{A}_\lambda(\mathcal{E}) := \{\bar{u} \in \bar{S} : \text{Alg. 7 retourniert ein } \bar{x} \in \mathcal{E} \text{ ausgehend von } \bar{u} \text{ mit Startschritt. } \lambda\}$$

Klarerweise gibt es ein  $\lambda > 0$ , so dass für ein lokales Grenzoptimum  $\bar{x}^*$  und eine Hochebene  $\mathcal{E}^+ \subset \bar{S}$  gilt:

$$\mathcal{A}_\lambda(\bar{x}^*) \supset \{\bar{x}^*\}$$

$$\mathcal{A}_\lambda(\mathcal{E}^+) \supset \{\mathcal{E}^+\}$$

Für eine Tiefebene  $\mathcal{E}^- \subset \bar{S}$  hält folgende Überlegung: Für alle  $\lambda > 0$  gilt:

$$\mathcal{A}_\lambda(\mathcal{E}^-) \subseteq \{\mathcal{E}^-\}$$

Für Zwischenebenen (so es sie überhaupt gibt) können wir keine klare Aussagen bezüglich der Teilmengenbeziehung treffen.

Das heißt: Bei geeigneter (dh nicht zu großen) Wahl der Startschrittweite werden über kurz oder lang sämtliche vorhandenen lokalen Grenzoptima aufgespürt. Falls es Hochebenen gibt, so erhalten wir darüber hinaus früher oder später eine Vielzahl ähnlicher Lösungen aus jeder Hochebene. Gegebenenfalls gesellen sich Lösungen aus Tief- und Zwischenebenen hinzu, so es welche gibt.

Wir erinnern uns an die Diskussion in Kapitel 5.4.2 ab Seite 93, wo wir die „richtige“ Wahl der Startschrittweite  $\lambda$  thematisiert haben. Wir sollten nicht in Panik geraten und eine winzig kleine Startschrittweite wählen. Das würde die Rechenzeit unnötig in die Höhe schmalzen. So hat sich die Startschrittweite  $\lambda = 0.0256$  als guter Startwert herausgestellt (für  $\sigma = 1$ ).

**Bemerkung 53:** Falls es Ebenen gibt, so liefert uns Algorithmus 9 also möglicherweise eine Vielzahl ähnlicher Lösungen. Im Anhang in Abschnitt 7.1 schneiden wir kurz an, wie wir mit Hilfe der Clusteranalyse die Fülle an Lösungen komprimieren können.

Auch wenn wir hier nicht nachweisen können, dass das Kompetenzentwicklungssystem für ein gegebenes Grenzportfolio  $\bar{x}$  gegen einen entsprechenden Fixpunkt konvergiert, so sind die mittels Algorithmus 9 gefundenen Grenzportfolios sehr aussichtsreiche Kandidaten für langfristig lokal optimale Lösungen.

**Beispiel 54:** Um das Kapitel würdig zu beenden, zeigen wir die Attraktionsbecken des Triathlonbeispiels für die übliche Konstellation. Zur Erinnerung sind die lokalen Grenzoptima nochmals in Tabelle 10 aufgelistet.

Punkt	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{w}(\bar{x})$	$\tilde{\mathcal{N}}$	$ \tilde{\mathcal{N}} $
A	0.3710	0.4624	0.1666	220.40	{1, 2, 3}	3
C	0.0000	0.8334	0.1666	183.34	{2, 3}	2
G	0.0000	0.0000	1.0000	100.00	{3}	1

Tabelle 10: Übersicht über alle lokalen Grenzoptima im Triathlonbeispiel

In Abbildung 26 sind die Attraktionsbecken dargestellt. Man erkennt sehr schön, dass sich die Grenze des Attraktionsbeckens des Punktes G deutlich im tiefblauen Bereich befindet. Und diese Grenze würde nach rechts unten wandern, wenn wir die Startschrittweite vergrößerten.

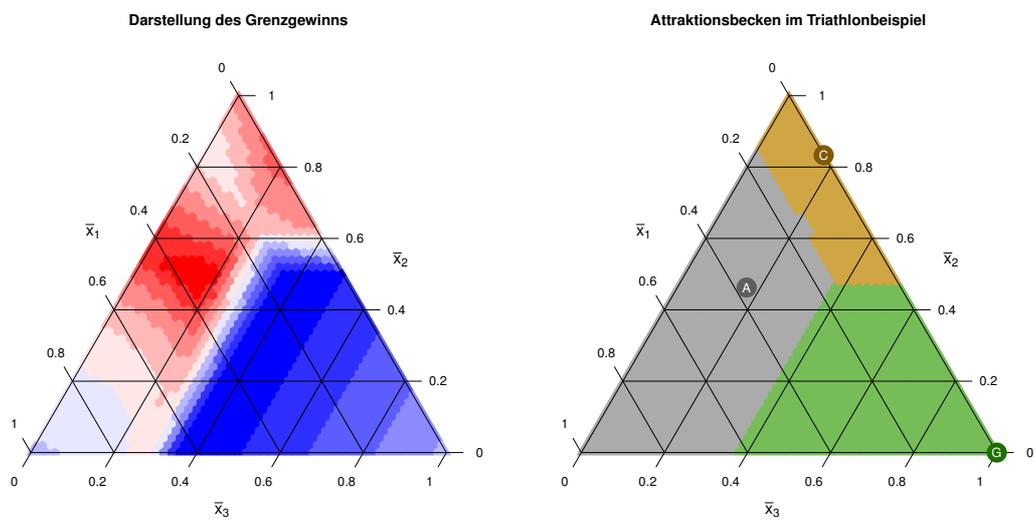


Abbildung 26: Attraktionsbecken im Triathlonbeispiel. **Links:** Darstellung der Grenzgewinne. **Rechts:** Die Attraktionsbecken der drei lokalen Grenzoptima für die Startschrittweite  $\lambda = 0.0256$ .

## 6. Mittelfristige lokale Optimierung

Mit diesem Kapitel schließen sich die Kreise dieser Arbeit. Wir werden beide Sichtweisen, die kurzfristige und die langfristige, jetzt unter einen Hut bringen. Nachdem wir in Abschnitt 6.2 erste Beispiele betrachtet haben, schreiben wir in Abschnitt 6.3 das Rolling Horizon Verfahren auf und werfen darauf folgend einen kurzen Blick auf die Parameterzusammenhänge sowie auf Gleichgewichtszustände.

### 6.1. Annahmen und Konventionen

Wir verwenden als Startlösung für den weiter unten aufgeschriebenen Algorithmus jene Lösungen, welche wir mit unserem Pattern Search Algorithmus 9 aus dem vorangehenden Kapitel erzeugt haben. Dort haben wir folgende Voraussetzungen verwendet:

$$\sigma_t = \sigma > 0 \quad \forall t \geq 1 \quad (126)$$

$$w_{it} = w_i > 0 \quad \forall t \geq 1, i \in \mathcal{N} \quad (127)$$

$$\alpha_r > 0 \quad \forall r \in \mathcal{R} \quad (128)$$

$$\beta_r > 0 \quad \forall r \in \mathcal{R} \quad (129)$$

Der in diesem Kapitel dargestellte Algorithmus kann auch ohne diese Annahmen sinnvoll angewendet werden. Ist die Abweichung zu den Annahmen allerdings groß, so machen die lokalen Grenzoptima als Startlösung kaum Sinn.

Falls es Ebenen gibt, so verweisen wir auf Abschnitt 7.1. Der Bequemlichkeit halber bezeichnen wir (die geclusterten) Punkte von Ebenen ebenfalls als lokale Grenzoptima.

### 6.2. Kompromiss zwischen kurzfristiger und langfristiger Optimierung

In Kapitel 4 haben wir die lokale Optimierung für  $T$  Perioden besprochen. Wir berechnen ausgehend von einer Startlösung ein Projektportfolio  $x$  derart, dass der abgezinste Ertrag

$$G(x) := \sum_{t \in \mathcal{T}} \delta^{-t} \sum_{i \in \mathcal{N}} w_{it} \eta_{it} x_{it} = \sum_{t=1}^T \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} \quad (130)$$

lokal maximiert wird. Knackpunkt dieses Vorgehens ist, dass es in keinster Weise langfristige Erträge berücksichtigt. Der gesamte Plan wird auf  $T$  Perioden ausgelegt, was danach passiert, tangiert uns nur periphär (fast so wie in der Politik ☺). Diese „Nach mir die Sintflut-Taktik“ kann also nicht der Weisheit letzter Schluss sein.

Unsere Gedanken schwenken zu Kapitel 5, in welchem wir den Grenzertrag

$$\bar{w}(\bar{x}) := \sum_{i=1}^n w_i \bar{\eta}_i \bar{x}_i \quad (131)$$

lokal maximiert haben. Wir führen also die auf diese Art und Weise vorgeschlagene Politik (das lokale Grenzoptimum) immer und immer wieder aus.

Allerdings geht diese Rechnung höchstens dann auf, wenn sich das Kompetenzentwicklungssystem bereits im Gleichgewicht befindet. Ist die Abweichung zum Fixpunkt anfänglich groß, so ist diese Vorgehensweise kurzfristig (und evtl. auch langfristig) suboptimal.

**Beispiel 55 („Radtop“):** Wir werden anhand des Triathlonbeispiels gerade erwähnten Umstand aufzeigen. Zur Anwendung kommt die übliche Parameterkonstellation:

$$d = \begin{bmatrix} 0.45 & 0.35 & 0.2 \\ 0 & 0.8 & 0.2 \\ 0 & 0 & 1 \end{bmatrix} \quad \beta = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix} \quad \alpha = \begin{pmatrix} 0.7 \\ 0.4 \\ 0.9 \end{pmatrix} \quad w = \begin{pmatrix} 300 \\ 200 \\ 100 \end{pmatrix} \quad \sigma = 1$$

Zusätzlich spezifizieren wir die Startwerte  $z_1$  und  $\gamma_1$  (ergeben sich automatisch aus  $z_1$ ) sowie den Zinssatz und die Parameter  $a$  und  $b$  der Lernfunktion  $\varphi(\cdot)$  wie folgt:

$$z_1 = \begin{pmatrix} -0.6 \\ 0.7 \\ -0.3 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0.05 \\ 0.97 \\ 0.18 \end{pmatrix} \quad \delta = 1.025 \quad a = 1 \quad b = 5$$

Zu Beginn sind wir also ein Virtuose auf dem Rad, im Wasser und zu Fuß stellen wir uns aber nicht allzu gut an<sup>5</sup>.

Eine Periode umfasst in diesem Beispiel (sowie in allen, die noch folgen werden) 3 Monate, wir messen also vierteljährlich. In Abbildung 27 auf der nächsten Seite vergleichen wir drei Varianten für jeweils 40 Perioden, also 10 Jahre lang.

In der 1. Variante setzen wir das globale Grenzoptimum ein. Die Erträge sind bescheiden: Bis die Schwimmleistung auf einem akzeptablen Level ist, sind Zins und Zinseszins bereits so hoch, dass sie uns einen guten Teil unserer Erträge wegfuttern.

In Variante 2 hingegen versuchen wir, die Schwimmleistung durch verstärkte Ausübung von Triathlon schneller zu verbessern. Interessant dabei: Laufen wird (abgesehen von der ersten Periode) bis zur 26. Periode komplett vernachlässigt. Zwischen den Perioden 8 und 25 sieht es so aus, als würden wir uns in Richtung Punkt B ( $\bar{x} = (0.44, 0.56, 0)^T$ ) entwickeln. Erst ab der 26. Periode beginnt sich das Ganze auf das globale Grenzoptimum hinzubewegen. Wir erkennen, dass dadurch die Laufleistung enorm profitiert.

Variante 3 schlägt vor, es gar nicht erst mit dem Schwimmen zu versuchen, diese Kompetenz also aufzugeben um die gewonnene Zeit anderen Disziplinen zukommen zu lassen. So versuchen wir, die Laufleistung gezielt zu steigern, um die Effizienz beim Duathlon zu erhöhen, was bereits nach wenigen Perioden Früchte trägt.

Unterm Strich sind wir mit Variante 3 sowohl kurz- als auch langfristig am erfolgreichsten, wie uns ein Blick auf den Zielfunktionsvergleich verrät.

---

<sup>5</sup>Kritische Stimmen werden jetzt sagen, dass die hier gewählten Spezifikationen unrealistisch sind. Wie kann jemand, der perfekt Fahrrad fahren kann so schlecht im Laufen sein. Tatsächlich stimme ich zu, aber die Einblicke, die wir dadurch (auch in den folgenden Beispielen) gewinnen werden, trösten uns über diesen Umstand hinweg!

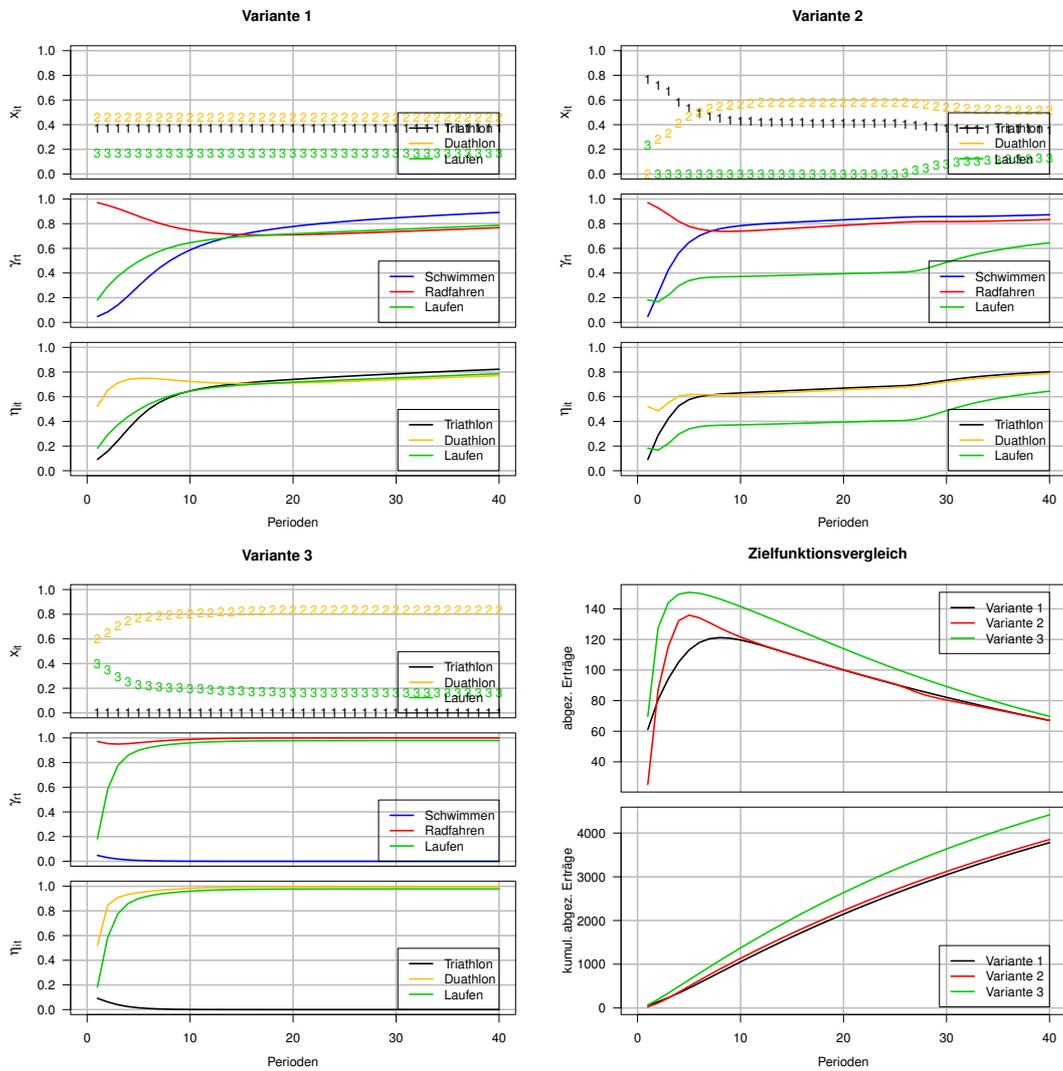


Abbildung 27: Vergleich dreier Lösungsvarianten in Beispiel „Radtop“. **Variante 1:** Wir setzen in jeder Periode das globale Grenzoptimum  $\bar{x} = (0.37, 0.46, 0.17)^T$  ein. **Varianten 2 & 3:** Zwei Vorschläge jenes Verfahrens, dem wir uns weiter unten widmen werden. **Zielfunktionsvergleich:** Die abgezinste sowie die kumulierten abgezinste Erträge aller Varianten auf einen Blick.

Wir betrachten ein zweites Beispiel.

**Beispiel 56 („Schwimmtop“):** *Wir modifizieren die Startwerte  $z_1$  gegenüber dem vorherigen Beispiel.*

$$z_1 = \begin{pmatrix} 0.5 \\ -0.1 \\ -0.4 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0.92 \\ 0.38 \\ 0.12 \end{pmatrix} \quad \delta = 1.025$$

*Hier stellen wir uns zu Beginn im Wasser sehr gut an. Auf dem Rad sind wir dagegen nur mäßig gut und zu Fuß sind wir nach wie vor schwach unterwegs.*

*In Abbildung 28 auf der nächsten Seite vergleichen wir erneut 3 Varianten.*

*In der 1. Variante setzen wir wiederum das globale Grenzoptimum ein. Im Gegensatz zum vorherigen Beispiel schneidet sie nicht schlecht ab, was an den Startwerten liegen dürfte. Würden wir zu Beginn zu viel in Laufen und Radfahren investieren, so nähme die Schwimmleistung ab. Und diese wieder aufzubauen, ist in dieser Konstellation schwer, da kein gezieltes Schwimmtraining möglich ist.*

*Die 2. Variante schlägt einen längerfristigen Mix aus Triathlon und Duathlon vor. Was wir nicht sehen: ab Periode 50 wird Laufen mit hineingemischt.*

*In der 3. Variante verbringen wir die ganze Zeit ausschließlich mit Triathlon. Diese Variante ist sehr kurzsichtig und tatsächlich wurde diese Lösung mit kurzsichtig eingestellten Parametern generiert.*

*Anfangs ist Variante 3 mit Abstand am vielversprechendsten. Nach rund 1.5 Jahren ist aber der abgezinste Ertrag erstmals kleiner als in Variante 2 und schon kurze Zeit später auch kleiner als in Variante 1. Wir erkennen, dass sich Variante 2 ab Periode 18 amortisiert.*

*Hier haben wir im Gegensatz zum vorherigen Beispiel eine Variante, die kurzfristig gut ist, aber von einer anderen Variante langfristig überholt wird. Das sture Einsetzen des globalen Grenzoptimums ist sowohl kurz- als auch langfristig suboptimal.*

Der Einfluss des Zinssatzes ist nicht zu unterschätzen. Wäre der Zinssatz niedriger, so würde sich im letzten Beispiel Variante 2 schon früher amortisieren.

Aber auch wenn sich das Kompetenzentwicklungssystem im zum globalen Grenzoptimum entsprechenden Gleichgewicht befindet, gibt es Lösungen, die kurzfristig besser sind. Sind anfänglich zB alle Kompetenzen perfekt entwickelt, so könnten wir kurzfristig mit Triathlon sehr viel ernten (300 pro Periode), wie wir in folgendem Beispiel erkennen.

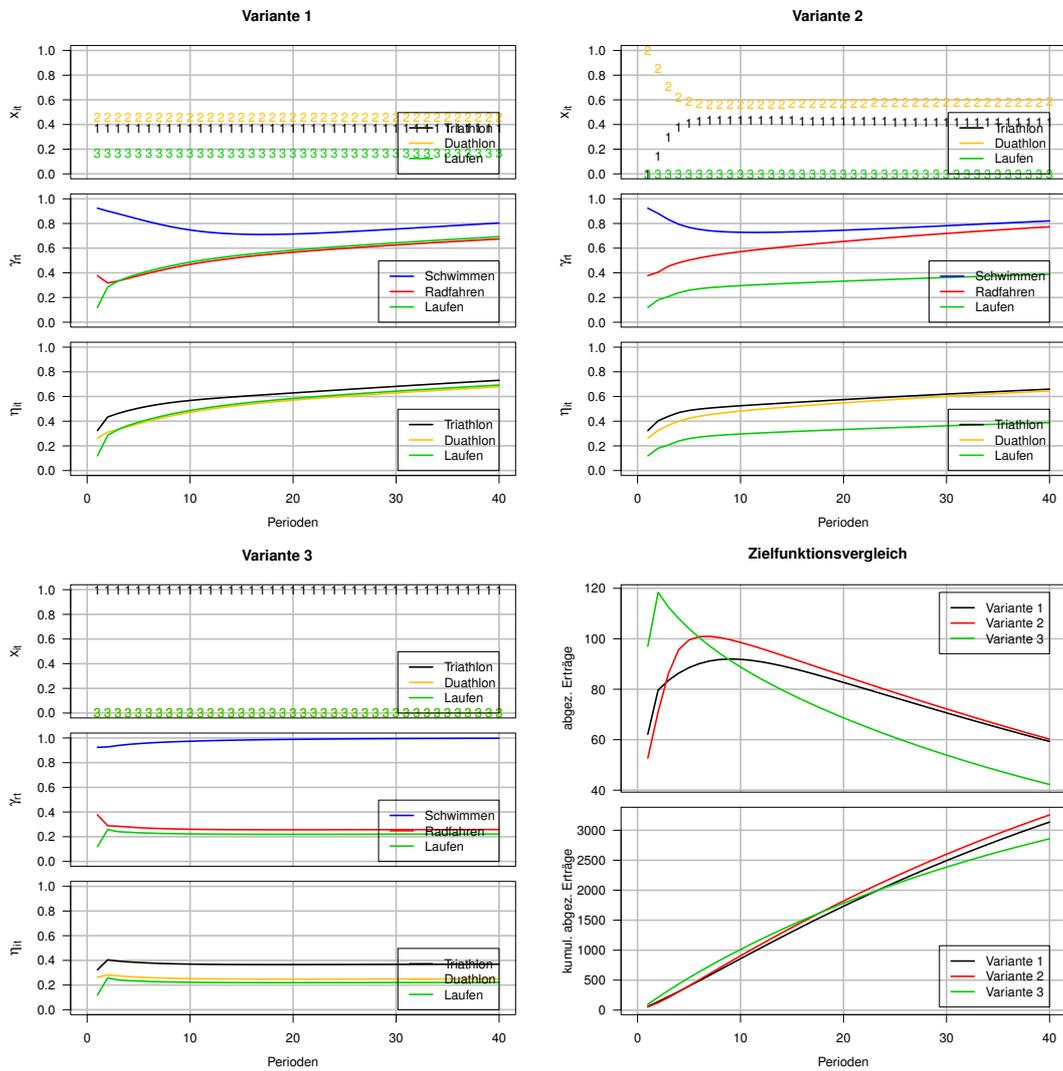


Abbildung 28: Vergleich dreier Lösungsvarianten in Beispiel „Schwimmtop“.

**Variante 1:** Wie im Beispiel zuvor setzen wir in jeder Periode das globale Grenzoptimum  $\bar{x} = (0.37, 0.46, 0.17)^T$  ein. **Varianten 2 & 3:** Zwei Vorschläge jenes Verfahrens, das wir weiter unten vorstellen und erläutern werden. **Zielfunktionsvergleich:** Die abgezinste sowie die kumulierten abgezinste Erträge aller Varianten auf einen Blick.

**Beispiel 57 („Perfektion“):** Wir verwenden die gleichen Daten wie in den vorherigen Beispielen, wobei wir den Zinssatz  $\delta$  etwas modifizieren und nahezu perfekte Startwerte für  $z_1$  und  $\gamma_1$  vorgeben:

$$z_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0.99 \\ 0.99 \\ 0.99 \end{pmatrix} \quad \delta = 1.0$$

Zusammengefasst: Anfangs in allen Disziplinen perfekt. Abbildung 29 auf der nächsten Seite gibt uns einen Überblick über einige mögliche Strategien.

Variante 1 ist wie gehabt das globale Grenzoptimum. Nichts Neues unter der Sonne.

Variante 2 ist die kurzfristige Triathlon-Pur-Lösung. Spontan fällt mir der Vergleich mit dem Fischfang ein. Zu Beginn ist der See noch voller Fische, doch schon bald wird der Bestand durch gierige Überfischung stark reduziert und der Ertrag fällt gnadenlos.

Variante 3 denkt etwas längerfristiger als Variante 2 (ist dabei aber immer noch recht kurzfristig eingestellt). Wir erkennen, dass zu Beginn der Triathlon voll ausgenützt wird. Ab der 6. Periode bemerken wir jedoch, dass wir den Verfall der Laufleistung durch gezieltes Training abbremsen können. Gleiches gilt ab Periode 12 – dort stoppen wir den Abfall der Radfahrleistung, indem wir Duathlon in den Mix aufnehmen.

Obwohl die Startkompetenzscores perfekt ins Schema des globalen Grenzoptimums passen, amortisiert sich Variante 1 gegenüber den beiden anderen erst ab der 30. Periode.

Wir halten unsere Beobachtungen fest:

- Stures Einsetzen lokaler Grenzoptima ist kurzfristig praktisch immer suboptimal.
- Stures Einsetzen lokaler Grenzoptima kann darüber hinaus auch mittel- oder sogar langfristig suboptimal sein, insbesondere dann, wenn sich die Startkompetenzwerte weit weg vom entsprechenden Fixpunkt befinden.

Wir müssen also einen Weg finden, beide Komponenten zu vereinen, die langfristige wie auch die kurzfristige. Das kann zum Beispiel mit Rolling Horizon passieren.

## 6.3. Bestimmung mittelfristiger lokaler Optima

### 6.3.1. Grundidee

Ausgehend von einer Startlösung führen wir für einen bestimmten Zeithorizont eine lokale Optimierung durch. Diese Politik führen wir aber nicht vollständig aus. Stattdessen korrigieren wir nach jeder Periode unseren Plan, indem wir für den selben Zeithorizont wiederum eine lokale Optimierung durchführen. Der **Horizontparameter**  $k > 0$  gibt dabei an, über wie viele Perioden sich jeweils die lokale Optimierung erstrecken soll.

Dabei fasst die Menge

$$\mathcal{T}_s^k := \{s, s + 1, \dots, s + k - 1\} \tag{132}$$

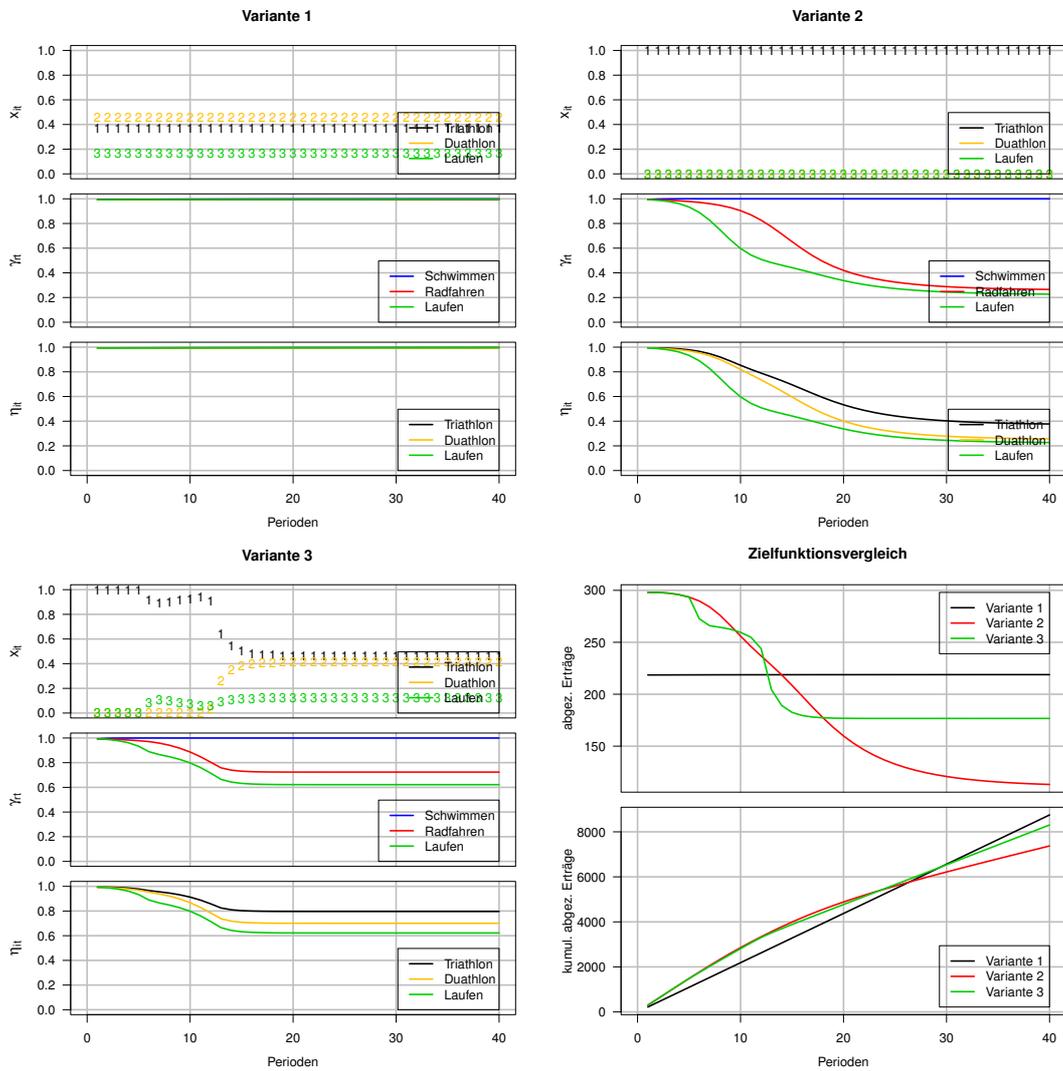


Abbildung 29: Vergleich dreier Lösungsvarianten in Beispiel „Perfektion“.

**Variante 1:** Wie gehabt setzen wir in jeder Periode das globale Grenzoptimum  $\bar{x} = (0.37, 0.46, 0.17)^T$  ein. **Varianten 2 & 3:** Zwei weitere Lösungsvorschläge jenes Verfahrens, das wir in Kürze vorstellen werden. **Zielfunktionsvergleich:** Die abgezinsten sowie die kumulierten abgezinsten Erträge aller Varianten auf einen Blick.

all jene Perioden zusammen, welche für die Berechnung des lokalen Optimums in der  $s$ . Periode relevant sind. Je größer der Horizontparameter  $k$ , desto mehr Perioden werden betrachtet.

Analog dazu definieren wir mit

$$G_s^k(x) := \sum_{t \in \mathcal{T}_s^k} \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} = \sum_{t=s}^{s+k-1} \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} \quad (133)$$

die dazu passende Zielfunktion. Mit dem Parameter  $T_1 > 0$  legen wir fest, wie oft wir die eben erwähnten Schritte wiederholen. Am Ende erhalten wir also ein Portfolio der Länge  $T_2 = T_1 + k - 1$ , wobei lediglich die ersten  $T_1$  Perioden zur Durchführen gelangen. Die letzten  $k - 1$  verwerfen wir.

Das ist die Grundidee des **Rolling Horizon Verfahrens**, welche wir in Abbildung 30 schematisch aufzeigen.

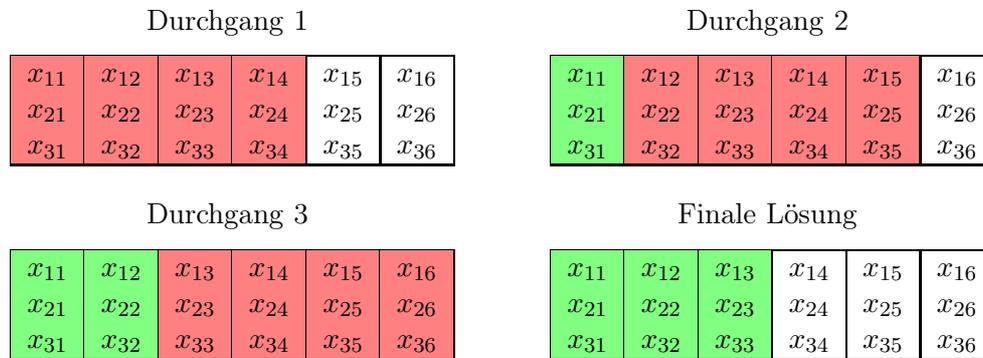


Abbildung 30: Schematische Darstellung des Rolling Horizon Verfahrens für  $k = 4$  und  $T_1 = 3$ . Wir beginnen damit, die ersten  $k = 4$  Perioden zu optimieren. Die erste Periode wird übernommen und die nächsten  $k$  Perioden optimiert. Das machen wir solange, bis wir die gewünschte Zahl an Perioden  $T_1 = 3$  bekommen. Die letzten  $k - 1 = 3$  Perioden sind irrelevant.

### 6.3.2. Wahl der Startlösung

Wie bringen wir am besten den Stein ins Rollen? Es gibt zahlreiche Möglichkeiten, die Startlösung zu wählen. So könnten wir beispielsweise Zufallslösungen einsetzen.

Wir werden hier aber eine andere Strategie verfolgen. Gemäß Algorithmus 9 generieren wir mit Iterated Local Search (fast) alle lokalen Grenzoptima und fassen diese in der Liste  $\mathcal{L}$  zusammen.

Sei  $\bar{x} \in \mathcal{L}$ . Dann setzen wir

$$x_{it} \leftarrow \bar{x}_i \quad \forall i \in \mathcal{N}, t \in \mathcal{T}_1^k \quad (134)$$

Dies könnten bzw. sollten wir sodann für alle  $\bar{x} \in \mathcal{L}$  tun, um eine breitere Palette an Lösungen zu erhalten.

Die Grundidee dahinter ist einfach: Wir setzen eine langfristig vielversprechende und stabile Lösung ein. Ist sie kurzfristig ein absoluter Unsinn, so wird diese durch die lokale Optimierung ausgebessert. Wenn aber diese Lösung auch kurzfristig brauchbar ist, so erfolgt nur eine geringfügige Adaptierung.

Wenn wir sämtliche lokalen Grenzoptima kennen würden, so bekämen wir für mehrere Werte für  $k$  einen sehr guten Überblick über die möglichen Strategien, derer zu bedienen wir imstande wären.

### 6.3.3. Aufrollen der Lösung

Angenommen wir haben den ersten Durchgang schon erfolgreich gemeistert. Was tun wir ab dem zweiten? Es gibt auch hier mehrere Möglichkeiten: Wir können wiederum den Zufall spielen lassen, oder weiterhin lokale Grenzoptima (oder Lösungen von Ebenen) einsetzen...

Wir werden uns hier aber auf das **Aufrollen der Lösung** konzentrieren. Dabei verwenden wir als Startlösung für Durchgang  $s$  ( $s \geq 2$ ) einfach die optimierte Lösung des vorherigen Durchgangs, also

$$x_{\mathcal{T}_s^k} \leftarrow x_{\mathcal{T}_{s-1}^k} \quad (135)$$

Dies stellt eine verkürzte Schreibweise für

$$x_{s+l-1} \leftarrow x_{s+l-2} \quad \forall l \in \{k, k-1, \dots, 1\} \quad (136)$$

dar. Dabei beachte man, dass man von hinten mit dem Aufrollen beginnen muss.

Diese Vorgehensweise hat einen entscheidenden Vorteil gegenüber allen anderen: Wenn das Kompetenzentwicklungssystem träge ist, so ändern sich die Kompetenzscores von einer Periode zur nächsten nicht allzu stark. Daher ist es legitim anzunehmen, dass sich die optimierte Lösung des vorherigen Durchgangs als Startwert für den aktuellen eignet. Es wird natürlich gewisse Anpassungen geben, aber die prinzipielle Struktur bleibt im Allgemeinen erhalten. Das spart Rechenzeit!

### 6.3.4. Rolling Horizon Algorithmus

In Algorithmus 10 auf der nächsten Seite schreiben wir den Rolling Horizon Algorithmus auf. Das Aufrollen erfolgt hier durch die Weiterverwendung von  $x^G$  in der nächsten Iteration. Dafür müssen die Kompetenzscores von der zweiten in die erste Periode kopiert werden.

---

**Algorithmus 10** Berechne mittelfristiges lokales Optimum – Rolling Horizon

---

**Input:**  $T_1 \in \mathbb{N}$  – gewünschte Periodenzahl,  $k \in \mathbb{N}$  – Horizontparameter,  $x^S$  – eine zulässige Lösung für  $k$  Perioden, die Startlösung

**Output:**  $x$  – eine  $n \times T_1$  Matrix gemäß Rolling Horizon

```
1: Initialisiere  $x$  als  $n \times T_1$  Matrix
2:  $x^C \leftarrow x^S$  // Startlösung der aktuellen Lösung (Current) zuweisen
3:  $x^C \leftarrow$  Berechne kurzfristiges lokales Optimum von  $G_1^k(x^C)$  gemäß Algorithmus 4
4:  $x_1 \leftarrow x_1^C$  // Kopiere erste Komponente in die Finallösung
5: for  $s = 2$  to  $T_1$  do
6:    $z_{r1} \leftarrow z_{r2} \quad \forall r \in \mathcal{R}$  // Kompetenzentwicklung
7:    $x^C \leftarrow$  Berechne kurzfristiges lokales Optimum von  $G(x^C)$  für  $T = k$  Perioden
   gemäß Algorithmus 4
8:    $x_s \leftarrow x_1^C$  // Kopiere erste Komponente in die Finallösung
9: end for
10: return  $x$ 
```

---

#### 6.4. Einfluss der Parameter $k$ , $\delta$ und $z_1$ sowie der Startlösung $x^S$

Wir halten fest:

- Die Berechnung der lokalen Grenzoptima erfolgt unabhängig von  $k$  und  $\delta$ . Da wir an einem Überblick über die langfristigen Entwicklungsmöglichkeiten interessiert sind, haben wir außerdem die Startkompetenzscores  $z_1$  außen vor gelassen.
- Im Gegensatz dazu fließen  $k$  und  $\delta$  sehr wohl in Rolling Horizon ein.
- Weiters bestimmen  $z_1$  und  $x^S$ , in welche Richtung sich die Lösung entwickelt.

Wir wenden uns nun dem zweiten Aspekt zu. Dazu zerlegen wir die Zielfunktion:

$$G_1^k(x) := \sum_{t=1}^k \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it} = \underbrace{\delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1}}_{\text{Gegenwart}} + \underbrace{\sum_{t=2}^k \delta^{-t} \sum_{i=1}^n w_{it} \eta_{it} x_{it}}_{\text{Zukunft}} \quad (137)$$

Schauen wir uns an, wie das Gewicht zwischen Gegenwart und Zukunft aussieht. Wir multiplizieren die Zielfunktion mit  $\delta$ :

$$\delta \cdot G_1^k(x) := 1 \cdot \underbrace{\sum_{i=1}^n w_{i1} \eta_{i1} x_{i1}}_{\text{Gegenwart}} + \underbrace{\sum_{t=2}^k \delta^{-t+1} \sum_{i=1}^n w_{it} \eta_{it} x_{it}}_{\text{Zukunft}} \quad (138)$$

Dann wird die Gegenwart mit 1 gewichtet. Das Gewicht der Zukunft wäre dann:

$$\sum_{t=2}^k \delta^{-t+1} = \sum_{t=1}^{k-1} \delta^{-t} = \delta^{-1} \cdot \frac{1 - \delta^{-(k-1)}}{1 - \delta^{-1}} \quad (139)$$

Bei den lokalen Grenzoptima haben wir die Gegenwart quasi komplett ausgeblendet und uns nur auf die Zukunft konzentriert.

Wir stellen einige **Folgerungen** an:

- Je größer der Horizontparameter  $k$ , desto mehr wiegt die Zukunft.
- Je größer der Zinssatz  $\delta$ , desto weniger wiegt die Zukunft. Wir versuchen, das Geld heute zu verdienen, weil es morgen (viel) weniger wert ist.

Tabelle 11 listet für ausgewählte  $k$  und  $\delta$  auf, um wie viel die Zukunft stärker gewichtet wird als die Gegenwart. Gemäß unseren Folgerungen von gerade eben ist die Chance, langfristig gute Lösungen zu erhalten umso größer, je mehr die Zukunft wert ist. Wie groß dieses Gewicht in der Praxis sein muss, um dies zu gewährleisten, hängt dann von der Parameterkonstellation ab.

$\delta \backslash k$	1	3	6	9	12	18	24	36	48	60	$\infty$
1	0	2	5	8	11	17	23	35	47	59	$\infty$
1.01	0	1.97	4.85	7.65	10.37	15.56	20.46	29.41	37.35	44.4	100
1.02	0	1.94	4.71	7.33	9.79	14.29	18.29	25	30.29	34.46	50
1.03	0	1.91	4.58	7.02	9.25	13.17	16.44	21.49	25.02	27.51	33.33
1.04	0	1.89	4.45	6.73	8.76	12.17	14.86	18.66	21.04	22.53	25
1.05	0	1.86	4.33	6.46	8.31	11.27	13.49	16.37	17.98	18.88	20
1.06	0	1.83	4.21	6.21	7.89	10.48	12.3	14.5	15.59	16.13	16.67
1.07	0	1.81	4.1	5.97	7.5	9.76	11.27	12.95	13.69	14.02	14.29
1.08	0	1.78	3.99	5.75	7.14	9.12	10.37	11.65	12.16	12.37	12.5
1.09	0	1.76	3.89	5.53	6.81	8.54	9.58	10.57	10.92	11.04	11.11
1.1	0	1.74	3.79	5.33	6.5	8.02	8.88	9.64	9.89	9.96	10

Tabelle 11: Gewicht der Zukunft für ausgewählte  $k$  und  $\delta$ . Die einzelnen Werte zeigen, um wieviel die Zukunft relativ gesehen mehr wiegt als die Gegenwart.

Wir werden uns diesen Zusammenhang und die genauen Einflüsse der Parameter in dieser Arbeit nicht näher ansehen. Zu kompliziert sind die Einflüsse, als dass eine Studie auf wenigen Seiten möglich wäre. Aber vielleicht habe ich Sie, werter Leserin und werter Leser dazu inspirieren, sich genauer diesem Thema zuzuwenden?

Stattdessen illustrieren wir unsere Überlegungen an Hand des Triathlonbeispiels.

**Beispiel 58 (Fortsetzung „Perfektion“):** *Wir rollen Beispiel „Perfektion“ auf Seite 108 nochmals auf und klären, wie die Ergebnisse zustande gekommen sind. Zur Anwendung gelangt die übliche Parameterkonstellation, sowie*

$$z_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0.99 \\ 0.99 \\ 0.99 \end{pmatrix} \quad \delta = 1.0 \quad a = 1 \quad b = 5$$

Abbildung 31 auf Seite 115 zeigt die Situation für 120 Perioden (also 30 Jahre).

**Beispiel 59 (Fortsetzung „Radtop“):** Wir betrachten jetzt nochmals Beispiel „Radtop“ auf Seite 104 und lüften das Geheimnis, wie die Lösungen in Variante 2 und 3 zustande gekommen sind. Die übliche Parameterkonstellation kommt zur Anwendung, sowie

$$z_1 = \begin{pmatrix} -0.6 \\ 0.7 \\ -0.3 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0.05 \\ 0.97 \\ 0.18 \end{pmatrix} \quad \delta = 1.025 \quad a = 1 \quad b = 5$$

Abbildung 32 auf der nächsten Seite veranschaulicht das Beispiel für 120 Perioden.

## 6.5. Gleichgewichte und Konvergenz

Uns fällt etwas auf: Die Lösungen streben gegen Mischungen, welche erstaunliche Ähnlichkeiten mit den besonderen Punkten A, B und C haben! Folgende Definition zeigt uns, wie ein stabiler Gleichgewichtszustand beim Rolling Horizon Verfahren aussieht.

**Definition 60:** Eine *lokal optimale k-Schablone*  $\hat{x} \in \mathbb{R}^{n \times k}$  ist eine zulässige Lösung für  $k$  Perioden, sodass folgende Eigenschaften erfüllt sind:

1.  $\gamma_{r2} = \gamma_{r1} \quad \forall r \in \mathcal{R}$
2.  $\hat{x}$  ist lokales Optimum bezüglich  $G_1^k(\cdot)$

Wir nennen sie auch einfach kurz **k-Schablone**.

Das heißt, die Effizienzen ändern sich von Periode 1 auf Periode 2 nicht, wenn wir  $\hat{x}_1$  einsetzen und ein lokales Optimum ist erreicht.

Auch wenn die Konvergenz in dieser Arbeit offen bleibt, so gibt es zumindest Indizien dafür, dass die Lösung des Rolling Horizon Verfahrens gegen eine k-Schablone strebt.

**Beispiel 61 (Fortsetzung „Radtop“):** Wir betrachten in Tabelle 12 auf Seite 116 zwei 12-Schablonen und vergleichen sie mit den Lösungen in Abbildung 32 auf Seite 115. In Variante 2 ist nach 120 Perioden (noch) eine Abweichung in der Größenordnung von etwa  $\pm 0.02$  festzustellen, während in Variante 3 die 12-Schablone praktisch abweichungsfrei erreicht wird.

Wenn die Lösung von Rolling Horizon gegen eine schlechte langfristige Lösungen strebt, so besteht die Gefahr eines langfristigen Absturzes des Ertrags, insbesondere dann, wenn der Zinssatz niedrig ist. So gesehen in den Beispielen 56 („Schwimmtop“) und 57 („Perfektion“) – Triathlon pur ist langfristig gesehen mau.

Wir werden abschließend in den Abbildungen 33 und 34 den Zusammenhang zwischen lokalen Grenzoptima und k-Schablonen andiskutieren.

Die hier dargestellten k-Schablonen wurden wie folgt generiert:

- Setze für ein lokales Grenzoptimum  $L(\bar{x}, \bar{\gamma}, \bar{\eta}, \bar{w})$  die Startkompetenzen  $\gamma_1 = \bar{\gamma}$  und setze  $x_t = \bar{x}$  für  $t \in \mathcal{T}_1^k$ .
- Wende Rolling Horizon hinreichend lange an.

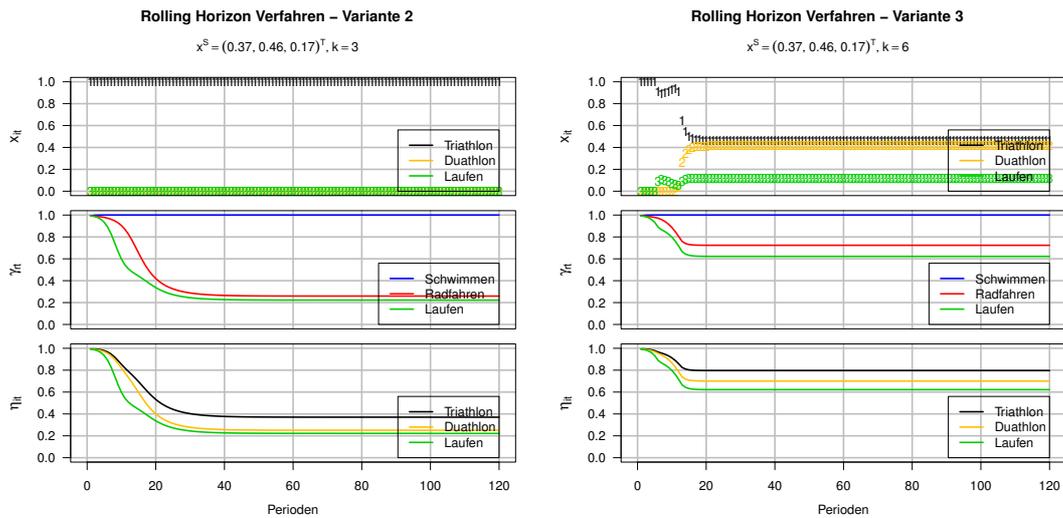


Abbildung 31: Vergleich zweier Lösungsvarianten in Beispiel „Perfektion“. Wir setzen in beiden Varianten das globale Grenzoptimum als Startlösung ein.  
**Links:** Für  $k = 3$  konvergiert das Verfahren noch zum „Triathlon total“.  
**Rechts:** Für  $k = 6$  kann das Verfahren schon längerfristig denken.

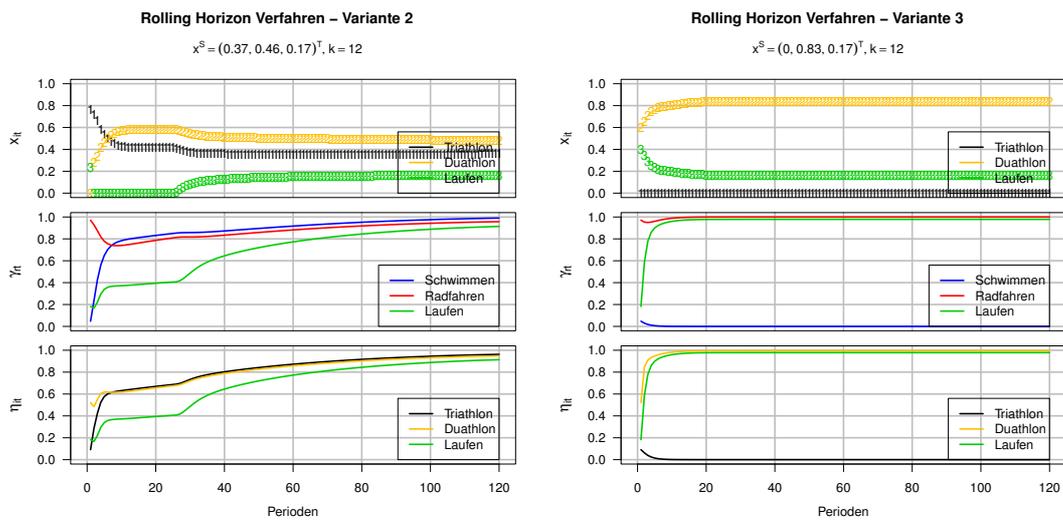


Abbildung 32: Vergleich zweier Lösungsvarianten in Beispiel „Radtop“. **Links:** Wir setzen das globale Grenzoptimum  $\bar{x} = (0.37, 0.46, 0.17)^T$  als Startlösung ein. Nach einer Art Sortierphase, beginnt sich etwa ab Periode 30 eine stabile Lösung herauszukristallisieren. **Rechts:** In dieser Variante setzen wir als Startlösung  $\bar{x} = (0, 0.83, 0.17)^T$  ein. Schon recht früh (etwa ab Periode 20) beginnt sich die Lösung einzupendeln.

$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$	$\hat{x}_6$	$\hat{x}_7$	$\hat{x}_8$	$\hat{x}_9$	$\hat{x}_{10}$	$\hat{x}_{11}$	$\hat{x}_{12}$
0.3799	0.6147	0.8972	0.9118	0.935	0.9738	1	1	1	1	1	1
0.4611	0.2485	0.0000	0.0000	0.000	0.0000	0	0	0	0	0	0
0.1590	0.1368	0.1028	0.0882	0.065	0.0262	0	0	0	0	0	0
0.000	0.0000	0.0000	0.0000	0.000	0.0000	0	0	0	0	0	0
0.837	0.8381	0.8406	0.8482	0.871	0.9352	1	1	1	1	1	1
0.163	0.1619	0.1594	0.1518	0.129	0.0648	0	0	0	0	0	0

Tabelle 12: 12-Schablonen des Triathlonbeispiels „Radtop“. **Oben:** 12-Schablone für Variante 2. Sie konzentriert sich sukzessive auf Triathlon. **Unten:** 12-Schablone für Variante 3. Sie konzentriert sich sukzessive auf Duathlon.

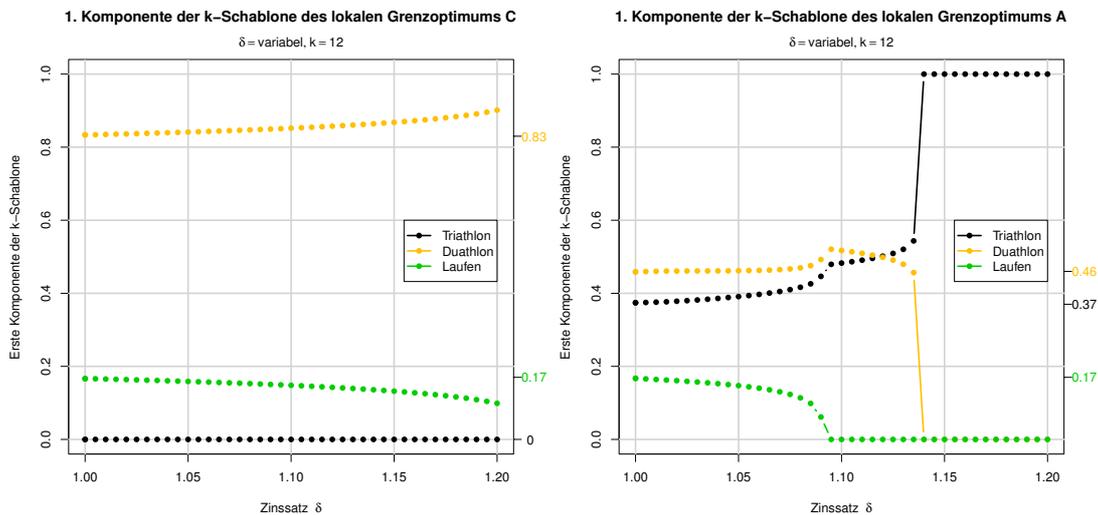


Abbildung 33: k-Schablonen im Triathlonbeispiel mit  $k = 12$  und variablem  $\delta$ .

**Links:** Die erste Komponente der 12-Schablone entfernt sich mit steigendem Zinssatz schleichend vom lokalen Grenzoptimum  $(0, 0.83, 0.17)^T$ , da sich die Investition in Laufen für große Zinssätze weniger lohnt.

**Rechts:** Der Anteil des Laufens sinkt stetig mit wachsendem  $\delta$ . Ab einem Zinssatz von knapp 1.095 ist es dann mit dem Laufen vorbei. Wir haben dann einen Zustand, der in etwa dem lokalen Grenzoptimum B entspricht. Bei  $\delta = 1.14$  bricht plötzlich der Duathlon weg.

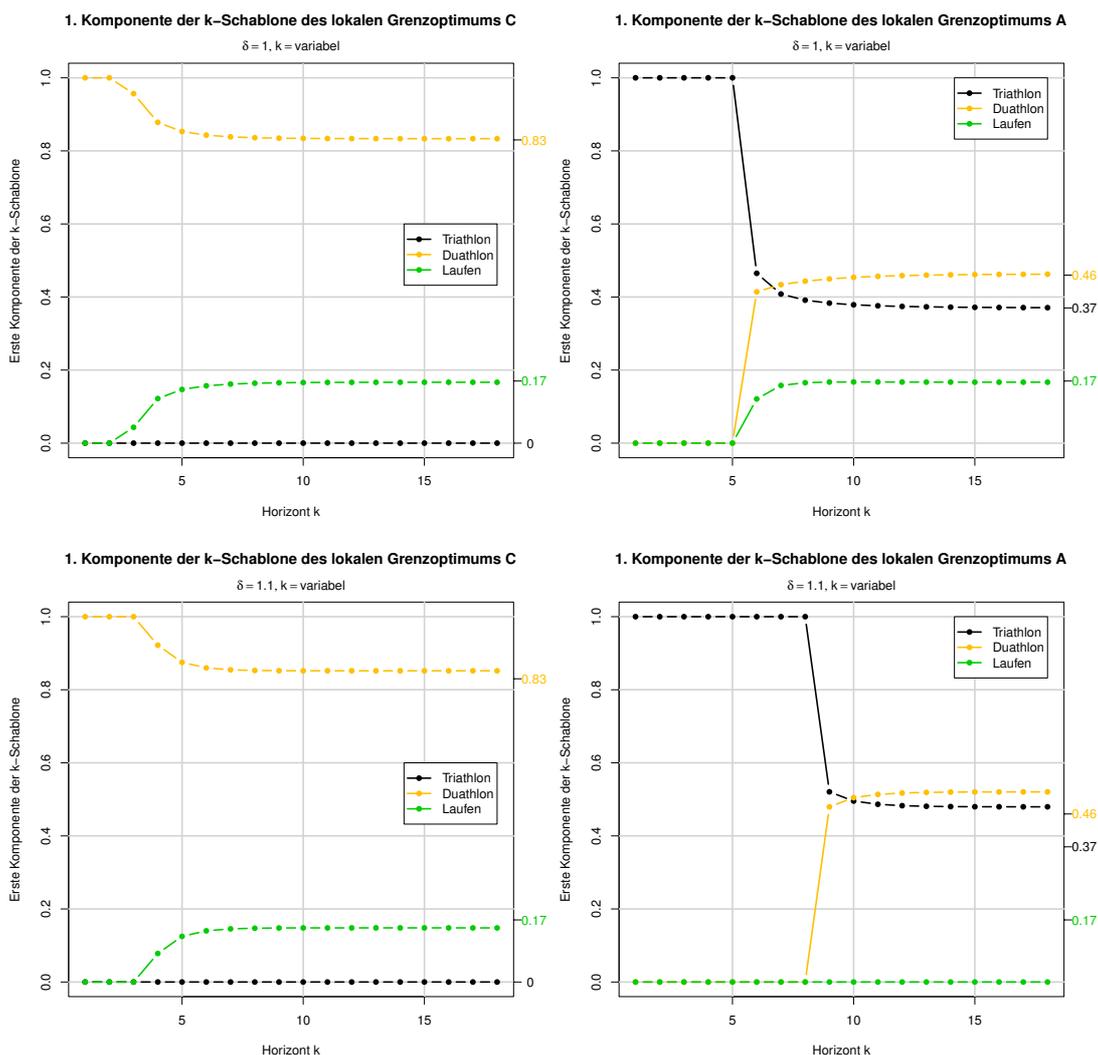


Abbildung 34:  $k$ -Schablonen im Triathlonbeispiel mit variablem  $k$ ,  $\delta = 1$  und  $\delta = 1.1$ .

**Links oben:** Für kleine  $k$  stürzen wir uns noch gierig auf den Duathlon, aber schon bald merken wir, dass sich Laufen hervorragend dazu eignet, die Effizienz von Duathlon zu erhöhen.

**Links unten:** Erhöhen wir ceteris paribus den Zinssatz, so entfernt sich die  $k$ -Schablone immer mehr vom lokalen Grenzoikum  $(0, 0.83, 0.17)^T$ .

**Rechts oben:** Anfangs stürzen wir uns noch gierig auf den Triathlon, erst bei  $k \geq 6$  wird sukzessive mehr Duathlon und Laufen hineingemischt.

**Rechts unten:** Erhöhen wir ceteris paribus  $\delta$ , so entfernt sich die  $k$ -Schablone zunehmend vom lokalen Grenzoikum. Ob des hohen Zinssatzes werden wir dazu genötigt, auf Laufen gänzlich zu verzichten und erst viel später als vorhin den Duathlon ins Portfolio mit aufzunehmen.

## 6.6. Abschließende Bemerkungen zur mittelfristigen Optimierung

Generell ist die Frage nach der Wahl des „richtigen“ Wertes für den Horizontparameter  $k$  nicht eindeutig beantwortbar. Es hängt davon ab, wie langfristig wir denken wollen (bzw. für Politiker, Manager: wie lange wir denken *dürfen*).

Am besten kann man diesen Umstand anhand des Marathons veranschaulichen. Sprinten wir zu Beginn mit Vollgas los, so wird uns schon nach kurzer Zeit die Kraft ausgehen. Wir brechen ein und werden von einer Horde intelligenterer Läufer überrollt (und dabei auch überholt). In diesem Fall haben wir  $k$  eindeutig zu klein eingestellt.

Wir könnten den Marathon aber auch im Schrittempo beginnen, damit wir der Gefahr des Einbruchs sicher entfliehen. Allerdings werden wir uns dann im Ziel über die großen Kraftreserven ärgern: „Hätte ich nur schneller begonnen, dann wäre ich nicht so weit hinten im Feld...“ Eindeutig haben wir  $k$  zu groß eingestellt.

Einige unerfahrene LäuferInnen würden sich in diesem Fall wohl darüber ärgern, der Marathon sei zu kurz gewesen. Die Distanz des Marathons wird (wohl) immer gleich bleiben und es liegt einzig und alleine an uns, unsere Kräfte bestmöglich einzuteilen.

Es muss also ein „optimales“  $k$  in der Mitte beider Extreme geben. Dieses zu finden erfordert eine Menge Fingerspitzengefühl und Erfahrung. Je besser wir die Situation einschätzen können, desto leichter tun wir uns beim Finden. Wie sollen wir beim Marathon die optimale Geschwindigkeit wählen, wenn wir noch nie einen absolviert haben?

Gleichzeitig schwingt aber immer die Gefahr von (plötzlichen) Preis- und Zinsänderungen mit, die teilweise nicht absehbar sind. Ein zu hoher Horizontparameter kann also dazu führen, Investitionen in eine Zukunft zu tätigen, die dann gar nicht so stattfinden wird. Und je ferner die Zukunft liegt, desto unberechenbarer wird sie.

Wie würden Sie, werte Leserin und werter Leser einen Laufbewerb angehen, wenn Sie noch nicht genau wissen, wie groß die Distanz ist? Würden Sie schneller loslaufen? Sich gleich ins vorderen Feld begeben, aus Angst, das Ziel könnte jederzeit auftauchen und die Gefahr des Einbruchs hinnehmen? Oder würden Sie langsamer loslaufen? Sich zu Beginn eher im hinteren Teil des Feldes aufhalten um stets über genügend Reserven zu verfügen aber mit der Gefahr, dass das Ziel schneller auftauchen könnte, als Ihnen lieb ist?

Wir werden dieses Thema kurz in Abschnitt 7.2 ansprechen. Und wir werden sehen, dass uns die stochastische Programmierung vor einer zu großen Wahl von  $k$  „beschützt“, da die Rechenzeit explosionsartig in  $k$  ansteigt.

Ach ja, eine Auflösung bin ich Ihnen noch schuldig! In Beispiel „Schwimmtop“ auf Seite 106 wurden folgende Parametereinstellungen verwendet:

**Variante 2:**  $k = 12$ ,  $x^S = (0.37, 0.46, 0.17)^T$

**Variante 3:**  $k = 3$ ,  $x^S = (0.37, 0.46, 0.17)^T$

## 7. Diskussion und Anregungen

Wir schneiden einige interessante Aspekte an, die den Rahmen dieser Magisterarbeit bei genauerer Betrachtung gesprengt hätten, bzw. sich nicht einordnen ließen. Vielleicht dienen sie nachfolgenden Studierenden als Inspiration für ihre Abschlussarbeiten...

### 7.1. Umgang mit Ebenen

In Abschnitt 5.4.3 haben wir drei Beispiele betrachtet, in denen es Ebenen gibt. Wir werden nun skizzieren, wie wir die Fülle an Lösungen verdichten können.

Alle von Algorithmus 9 gefundenen Lösungen werden derart in Töpfe unterteilt, dass jeder Topf aus (unterscheidbaren) Lösungen gleichen Grenzgewinns besteht.

Jetzt verwenden wir die Clusteranalyse (siehe zB [Seb04]), um die Lösungen jedes Topfes zu gruppieren. Wir benötigen hierzu ein Maß für die Unähnlichkeit zweier Grenzportfolios. Wie verwenden in den beiden Beispielen unten die Manhattanndistanz:

$$d(\bar{x}^1, \bar{x}^2) = \sum_{i=1}^n |\bar{x}_i^1 - \bar{x}_i^2|$$

Weiters verwenden wir die **hierarchische Clusteranalyse mit average linkage**. Das heißt, zu Beginn stellt jede Lösung ein eigenes Cluster dar. Die Distanz zweier Cluster ist definiert als das Mittel über alle paarweisen Distanzen, also

$$d(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{\bar{x}^1 \in C_1} \sum_{\bar{x}^2 \in C_2} d(\bar{x}^1, \bar{x}^2)$$

Nun werden schrittweise immer jene zwei Cluster zusammengefasst, welche die geringste Distanz zueinander aufweisen, solange bis nur noch ein Cluster übrig ist.

**Wichtiges Detail:** Vor der Clusteranalyse scheiden wir alle Lösungen aus, die im Laufe der lokalen Optimierung nicht verbessert werden konnten! Das bringt folgende Vorteile:

- Lösungen aus Tiefebenen werden eliminiert.
- Da die Lösungsdichte am Rand von Hochebenen deutlich größer als im Inneren ist, könnten uns Punkte im Inneren beim Clustern stark irritieren.

**Beispiel 62 (Fortsetzung 2. Ebenenbeispiel):** *Wir betrachten nochmals das 2. Ebenenbeispiel von Seite 96 und veranschaulichen es in Abbildung 35 auf der nächsten Seite.*

$$d = \begin{bmatrix} 0 & 1 \\ 3/4 & 1/4 \end{bmatrix} \quad \beta = \begin{pmatrix} 1/5 \\ 1/5 \end{pmatrix} \quad \alpha = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \quad w = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad \sigma = 1$$

**Beispiel 63 (Fortsetzung 3. Ebenenbeispiel):** *Wir rollen nochmals Variante 1 des 3. Ebenenbeispiels auf Seite 97 auf und stellen es in Abbildung 36 auf Seite 121 dar.*

$$d = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0 & 0.9 & 0.1 \\ 0.1 & 0 & 0.9 \end{bmatrix} \quad \beta = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} \quad \alpha = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad w = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \sigma = 1$$

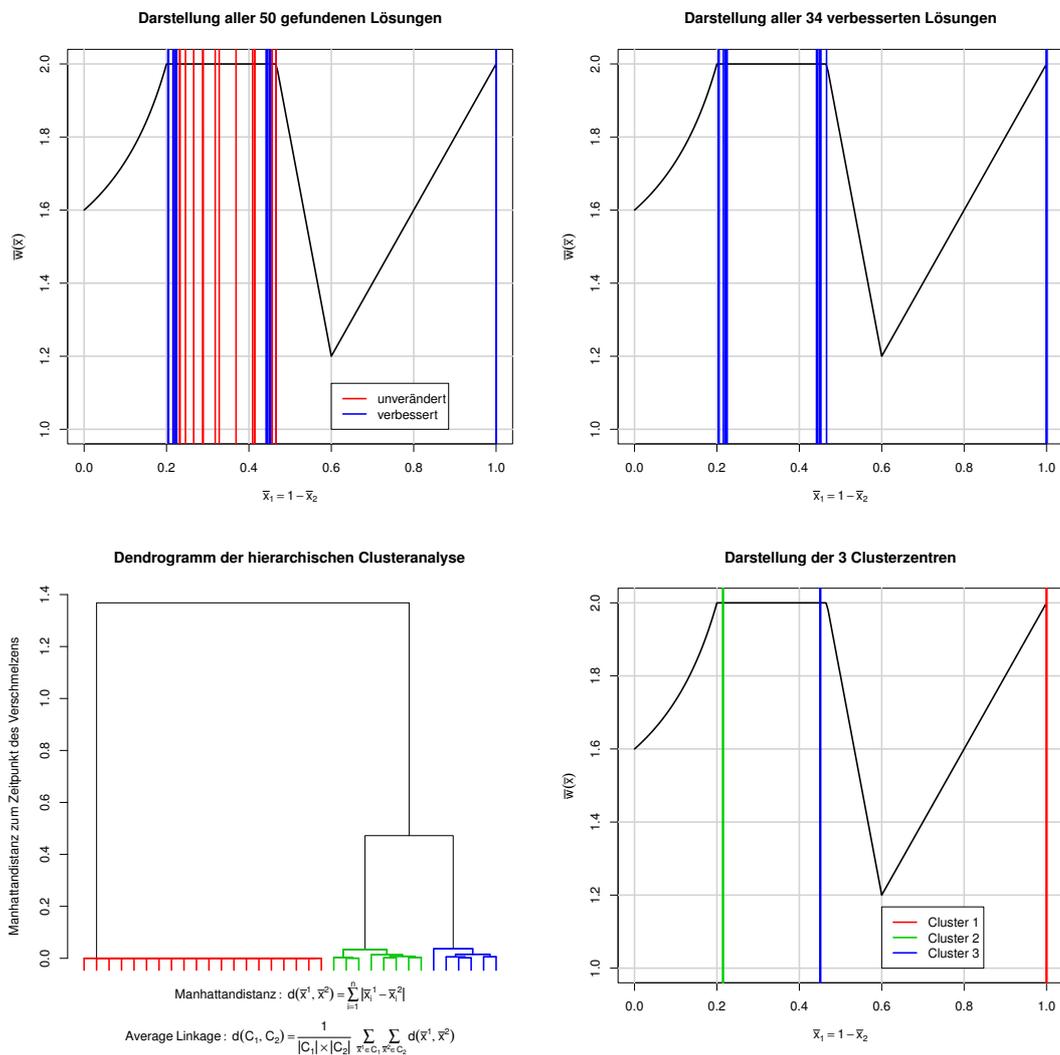


Abbildung 35: Darstellung der Clusteranalyse im 2. Ebenenbeispiel. **Oben:** Darstellung von 50 lokal optimierten Lösungen. Die 16 rot markierten Lösungen sind von Pattern Search nicht verändert worden und werden daher ausgeschieden. **Unten links:** Das Dendrogramm der hierarchischen Clusteranalyse mit den 34 verbesserten Lösungen zeigt, wie unähnlich sich die Cluster bei ihrer Verschmelzung waren. Man sieht, dass 2 oder 3 Cluster vernünftig sind. **Unten rechts:** Wir entscheiden uns für drei Cluster (im Zweifel lieber eines mehr als weniger).

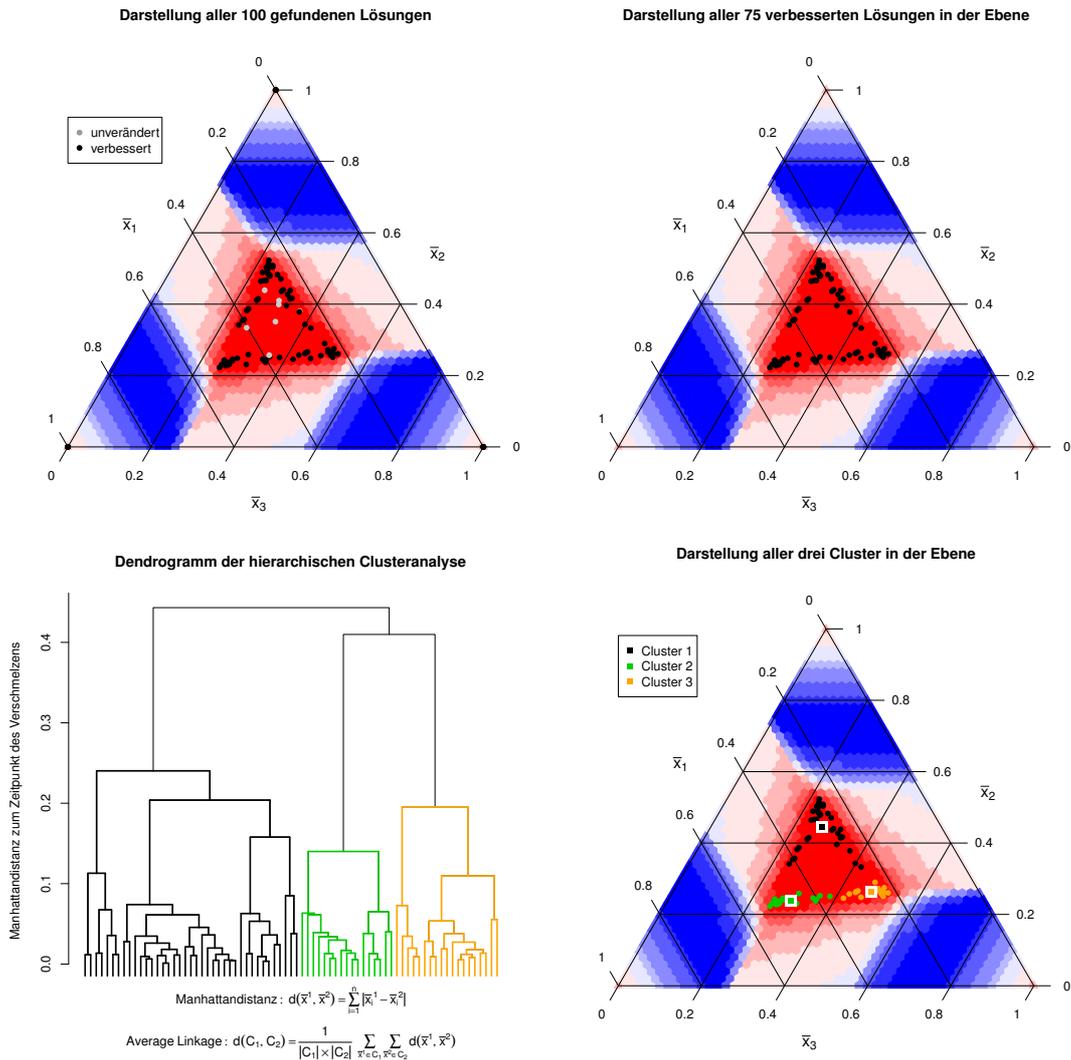


Abbildung 36: Darstellung der Clusteranalyse im 3. Ebenenbeispiel. **Oben:** Darstellung von 100 lokal optimierten Lösungen. Die 7 grau markierten Lösungen sind von Pattern Search nicht verändert worden und werden daher ausgeschieden. Die 18 Lösungen an den Ecken blenden wir für diese Clusteranalyse aus. **Unten links:** Das Dendrogramm der hierarchischen Clusteranalyse mit den 75 verbesserten Lösungen in der Ebene zeigt, wie unähnlich sich die Cluster bei ihrer Verschmelzung waren. Man sieht, dass 3 Cluster vernünftig sind. **Unten rechts:** Wir entscheiden uns also für drei Cluster. Die Quadrate markieren die jeweiligen Clusterzentren.

## 7.2. Unsichere Projekterträge und Zinssätze

In dieser Magisterarbeit haben wir einen fixen Zinssatz  $\delta$  unterstellt. Weiters waren die Projektklassenerträge  $w_{it}$  als deterministisch, in (bzw. ab) Kapitel 5 überdies als konstant vorausgesetzt.

Die tollen Zusammenhänge und Erkenntnisse, die wir gesammelt haben, können uns nicht davon hinwegtäuschen, dass diese Annahmen oftmals unrealistisch sind. Wirtschaftskrisen, wie jene aus dem Jahre 2008 lassen den Zinssatz im Nu in den Keller rasseln, was es von jetzt auf gleich viel attraktiver macht, mehr in die Zukunft zu investieren. Und auch die Erträge  $w_{it}$  unterliegen gewissen markttechnischen Mechanismen (Stichwort Angebot und Nachfrage).

### 7.2.1. Stochastisches Programm

Es besteht die Möglichkeit, diese Unsicherheiten zu modellieren. Stochastische Prozesse wie etwa der Wiener Prozess können Zinssatztrajektorien, sowie Szenarien für Preisschwankungen bei den Erträgen erzeugen.

Konkret seien  $W_{it}$  die Zufallsvariablen, welche die Erträge modellieren. Sei  $U_t$  die entsprechende Zufallsvariable für den Zinssatz in Periode  $t$ , wobei  $U_1 = \delta$  fast sicher gilt. Dann wäre das folgende **stochastische Programm** zu lösen:

$$\max_{x_1} \delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \mathbb{E} \left( \max_{x_2, \dots, x_k} \sum_{t=2}^k D_t \sum_{i=1}^n W_{it} \eta_{it} x_{it} \right) \quad (\text{SP})$$

wobei

$$D_t := \prod_{s=1}^t \frac{1}{U_s} \quad (140)$$

ist und die gleichen Nebenbedingungen gelten, wie im Basismodell.

Das stochastische Programm besteht aus einem deterministischen Teil – die Daten der ersten Periode sind bekannt, und einem stochastischen Teil – alle zukünftigen Perioden. Oftmals müssen wir uns mit lokalen Maxima begnügen.

Sei  $\Omega$  die Menge aller Szenarien. Jedes  $\omega \in \Omega$  enthält die Informationen darüber, wie sich der Zinssatz und die Erträge jeder Projektklasse bis Periode  $k$  entwickeln.

### 7.2.2. Der diskrete Fall

Wir betrachten den Fall, dass es insgesamt nur endlich viele Szenarien gibt,  $Q$  an der Zahl. Somit befinden sich  $Q$  Elemente in der Menge  $\Omega$ .

**Beispiel 64:** In jeder Periode gibt es 3 Möglichkeiten für die Entwicklung des Zinssatzes: Er kann steigen, gleich bleiben oder fallen, zB nach folgendem Schema:

$$U_{t+1} = \begin{cases} U_t \cdot c & \text{mit Wahrscheinlichkeit } 1/4 \\ U_t & \text{mit Wahrscheinlichkeit } 1/2 \\ U_t \cdot 1/c & \text{mit Wahrscheinlichkeit } 1/4 \end{cases} \quad \text{wobei} \quad U_1 = \delta \quad \text{fast sicher}$$

Dabei ist  $c \in \mathbb{R}$  eine Konstante. Analog gibt es die selben drei Möglichkeiten für die Entwicklung der Erträge jeder Projektklasse  $i$ :

$$W_{it+1} = \begin{cases} W_{it} + v_i & \text{mit Wahrscheinlichkeit } 1/4 \\ W_{it} & \text{mit Wahrscheinlichkeit } 1/2 \\ W_{it} - v_i & \text{mit Wahrscheinlichkeit } 1/4 \end{cases} \quad \text{wobei} \quad W_{i1} = w_{i1} \quad \text{fast sicher}$$

Die Konstanten  $v_i$  können von Projektklasse zu Projektklasse variieren.

Wir nehmen an, dass sich der Zinssatz und die Erträge der Projektklassen jeweils unabhängig voneinander entwickeln.

Somit folgt:

$$Q = |\Omega| = 3^{(n+1)(k-1)}$$

Das stochastische Programm im diskreten Fall sieht dann so aus:

$$\max_{x_1} \delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \sum_{q=1}^Q P(\omega_q) \max_{x_2, \dots, x_k} \sum_{t=2}^k D_t(\omega_q) \sum_{i=1}^n W_{it}(\omega_q) \eta_{it} x_{it} \quad (\text{SPD})$$

Dabei bezeichnet  $P(\omega_q)$  die Wahrscheinlichkeit, dass Szenario  $\omega_q$  eintritt.  $D_t(\omega_q)$  ist die Ausprägung für  $D_t$  in Szenario  $\omega_q$  und respektive  $W_{it}(\omega_q)$  jene für  $W_{it}$ .

Sofern  $Q$  nicht zu groß ist, kann (SPD) nach folgendem Prinzip gelöst werden: Gegeben  $x_1$  berechnen wir für jedes Szenario  $\omega_q \in \Omega$  die (lokal) optimale Politik  $x_2^*(\omega_q), \dots, x_k^*(\omega_q)$ . Dies ist mit den Techniken aus Kapitel 4 möglich.

Wir erhalten sogleich den erwarteten Zielfunktionswert gegeben  $x_1$ :

$$\delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \sum_{q=1}^Q P(\omega_q) \sum_{t=2}^k D_t(\omega_q) \sum_{i=1}^n W_{it}(\omega_q) \eta_{it}(\omega_q) x_{it}^*(\omega_q)$$

Hier bezeichnet  $\eta_{it}(\omega_q)$  die Effizienz von Projektklasse  $i$  in Periode  $t$  unter Anwendung der (lokal) optimalen Politik  $x_2^*(\omega_q), \dots, x_k^*(\omega_q)$  gegeben  $x_1$ .

Jetzt versuchen wir, jenes  $x_1$  zu finden, das diesen erwarteten Zielfunktionswert (lokal) maximiert. Hierfür können wir zB Pattern Search aus Kapitel 5 verwenden.

### 7.2.3. Der kontinuierliche Fall

Wenden wir uns jetzt dem Fall zu, dass es (überabzählbar) unendlich viele Szenarien gibt. In diesem Fall wäre in (SP) ein kompliziertes Integral zu lösen – an eine exakte Lösung ist daher kaum zu denken!

Ein Griff in die Statistik-Trickkiste hilft uns, (SP) hinreichend genau lösen zu können.

#### Lemma 65 (Starkes Gesetz der Großen Zahlen von Kolmogorov):

Sei  $X_1, X_2, X_3, \dots$  eine Folge unabhängig identisch verteilter Zufallsvariablen mit endlichem Erwartungswert  $\mathbb{E}(X_i) = \mu \in \mathbb{R}$ . Sei

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i \quad (141)$$

die Zufallsvariable des arithmetischen Mittels der ersten  $n$  Folgenglieder. Dann gilt:

$$\bar{X}_n \xrightarrow[n \rightarrow \infty]{} \mu \quad \text{fast sicher} \quad (142)$$

*Beweis.* Einen Beweis für den Spezialfall mit  $\mathbb{E}(X^4) < \infty$  findet man zB in [Pöt08].  $\square$

Angewendet auf unser stochastisches Programm fassen wir den Ausdruck

$$F_{x_1} = \max_{x_2, \dots, x_k} \sum_{t=2}^k D_t \sum_{i=1}^n W_{it} \eta_{it} x_{it} \quad (143)$$

als Zufallsvariable auf. Wir ziehen unter allen möglichen Szenarien  $\Omega$  eine Stichprobe der Größe  $Q$  heraus. Die einzelnen Szenarien beschriften wir mit  $\omega_q$ . Sie weisen alle die gleiche Eintrittswahrscheinlichkeit auf, es gilt also:

$$P(\omega_q) = \frac{1}{Q} \quad \forall q = 1, \dots, Q \quad (144)$$

Nun bezeichnet

$$f_q(x_1) := F_{x_1}(\omega_q) = \max_{x_2, \dots, x_k} \sum_{t=2}^k D_t(\omega_q) \sum_{i=1}^n W_{it}(\omega_q) \eta_{it} x_{it} \quad (145)$$

den (lokal) optimalen Wert, den man im stochastischen Teil von (SP) in Szenario  $\omega_q$  gegeben  $x_1$  erreichen kann. Lemma 65 angewandt ergibt

$$\frac{1}{Q} \sum_{q=1}^Q f_q(x_1) \xrightarrow[Q \rightarrow \infty]{} \mathbb{E}(F_{x_1}) = \mathbb{E} \left( \max_{x_2, \dots, x_k} \sum_{t=2}^k D_t \sum_{i=1}^n W_{it} \eta_{it} x_{it} \right) \quad \text{fast sicher} \quad (146)$$

und folglich

$$\delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \frac{1}{Q} \sum_{q=1}^Q f_q(x_1) \xrightarrow[Q \rightarrow \infty]{} \delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \mathbb{E}(F_{x_1}) \quad \text{fast sicher} \quad (147)$$

Das heißt, wir können die Lösung von (SP) hinreichend genau durch die Lösung von (SPD) annähern, sofern  $Q$  groß genug ist. Dabei sollte die Stichprobengröße  $Q$  exponentiell mit  $k$  wachsen, um eine hinreichend gute Samplingdichte zu erhalten. Wir geraten also leider früher als uns lieb ist an unsere Leistungsgrenze (bzw. an jene des Computers).

Entweder finden wir geeignete Importance Sampling Strategien, mit welchen wir die erforderliche Stichprobengröße durch intelligente Stichprobenziehung (stark) verringern können, oder wir verwenden ein ausgefeilteres Verfahren der stochastischen Optimierung, oder wir begnügen uns zB mit dem nächsten Unterabschnitt.

#### 7.2.4. Erwartetes Problem und der Value of Stochastic Solution

Viel leichter wird (SP) zu lösen, wenn wir Erwartungswert und max vertauschen. Unter der wichtigen Annahme, dass die Zufallsvariablen  $W_{it}$  und  $U_t$  paarweise unabhängig sind, gestaltet sich das dadurch gewonnene **erwartete Problem** wie folgt:

$$\begin{aligned} & \max_{x_1} \delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \max_{x_2, \dots, x_k} \mathbb{E} \left( \sum_{t=2}^k D_t \sum_{i=1}^n W_{it} \eta_{it} x_{it} \right) \\ &= \max_{x_1} \delta^{-1} \sum_{i=1}^n w_{i1} \eta_{i1} x_{i1} + \max_{x_2, \dots, x_k} \sum_{t=2}^k \mathbb{E}(D_t) \sum_{i=1}^n \mathbb{E}(W_{it}) \eta_{it} x_{it} \\ &= \max_x \sum_{t=1}^k \mathbb{E}(D_t) \sum_{i=1}^n \mathbb{E}(W_{it}) \eta_{it} x_{it} \end{aligned} \quad (\text{EP})$$

Auch hier gelten die Nebenbedingungen des Basismodells. Das Problem (EP) können wir leicht und schnell mit den Techniken aus Kapitel 4 lösen.

Allerdings wird jedweder Zufall „abgeschaltet“, weil wir lediglich das Durchschnittsszenario (lokal) optimieren. Dadurch erhalten wir eine Lösung, die idR suboptimal ist.

Der Schaden, welcher durch diese starke Vereinfachung entsteht, lässt sich mit folgender Definition beschreiben.

**Definition 66:** *Der Value of Stochastic Solution (kurz VSS) ist die Differenz zwischen der Lösung von (SP) und jene von (EP).*

Je größer VSS, desto eher sollte man sich die Mühe machen, das stochastische Programm zu lösen. Leider ist es schwierig, allgemeine Regeln dafür zu finden, wann VSS groß ist und wann nicht.

Daher versucht man in der Praxis, den VSS durch geeignete Schranken abzuschätzen. Weiters spielen neben dem VSS auch andere Maße eine Rolle, zB der EVPI (Expected Value of Perfect Information). Für eine ausführliche Einführung sei zB auf [BL97] verwiesen.

### 7.3. Hinarbeiten auf einen bestimmten Moment

Mit dem hier vorgestellten Modell ist es möglich, die Effizienzwerte  $\gamma_{rt}$  so zu entwickeln, dass in einer bestimmten Periode  $T$  ein  $\eta_{iT}$  möglichst groß ist. Das folgende Beispiel zeigt, wie.

**Beispiel 67:** *In 4 Monaten (bzw. bei monatlicher Betrachtung zu Beginn der 5. Periode) findet die Weltmeisterschaft im Triathlon statt. Wir wollen uns im Vorfeld so vorbereiten, dass wir genau zu dieser Zeit in Topform sind. Unser Ziel ist also:*

$$\eta_{15} \rightarrow \max! \quad (148)$$

*Da sich die Effizienzwerte immer auf den Periodenbeginn beziehen, müssen wir  $\eta_{15}$  und nicht  $\eta_{14}$  maximieren.*

*Dieses Vorhaben können wir leicht realisieren, indem wir die Erträge wie folgt festlegen<sup>6</sup>:*

$$w = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

*Das heißt, die einzige Möglichkeit an Erträge zu kommen, besteht darin, in der 5. Periode in Triathlon zu investieren. Werfen wir für  $\delta = 1$  einen Blick auf die Zielfunktion.*

$$G(x) = \sum_{t=1}^5 \sum_{i=1}^3 w_{it} \eta_{it} x_{it} = \eta_{15} x_{15} \rightarrow \max! \quad (149)$$

*Unsere Zielfunktion vereinfacht sich also zu  $\eta_{15} x_{15}$ . Klarerweise gilt  $x_{15} = \sigma_5$  in jedem lokalen Optimum und es wird in den ersten 4 Perioden alles getan, um  $\eta_{15}$  zu maximieren.*

Diese Idee lässt sich auf komplexere Fälle ausdehnen.

### 7.4. Korrelation zwischen den Projektklassen stärker einbauen

Wir haben in Beispiel „Radtop“ auf Seite 104 bemängelt, dass es unrealistisch ist, dass ein Radfahrer von Weltklasse nicht laufen kann. Auch in den folgenden Beispielen von Kapitel 6 können wir ähnliche Beanstandungen fällen.

---

<sup>6</sup>Technisch gesehen müssen wir, um den Anforderungen von Kapitel 4 zu genügen,  $\epsilon_0$  statt 0 schreiben. Dabei ist  $\epsilon_0 > 0$  eine winzig kleine Zahl, die sicherstellt, dass  $w_{it} > 0$  gilt. So ist es aber übersichtlicher.

Selbst wenn wir realistischere Startwerte  $z_1$  wählen würden, entwickelten sich die Kompetenzen Laufen und Radfahren praktisch unabhängig voneinander. Bzw. besteht die einzige Abhängigkeit beider Disziplinen einzig und allein darin, wie stark sie in den Sportarten Triathlon, Duathlon und Laufen vorkommen.

Im folgenden Beispiel skizzieren wir eine Möglichkeit, die Korrelation zwischen den Kompetenzen stärker berücksichtigen zu können.

**Beispiel 68:** *Die Disziplinen Schwimmen, Radfahren und Laufen bedürfen ihrerseits Kompetenzen. Um ein guter Läufer oder Radfahrer zu sein, benötigt man unter anderem Ausdauer und Beinkraft, als Schwimmer ist auch die Armkraft essentiell.*

*Wir erheben Schwimmen, Radfahren und Laufen zu **Projektklassen zweiter Ordnung** und fügen als dritte Stufe die Kompetenzen Ausdauer, Beinkraft und Armkraft ein:*

**Projektklassen 1. Ordnung:**  $i = 1$ : Triathlon,  $i = 2$ : Duathlon und  $i = 3$ : Laufen

**Projektklassen 2. Ordnung:**  $j = 1$ : Schwimmen,  $j = 2$ : Radfahren und  $j = 3$ : Laufen

**Kompetenzen:**  $r = 1$ : Ausdauer,  $r = 2$ : Armkraft und  $r = 3$ : Beinkraft

Allgemein könnte das Setting wie folgt aussehen:

$\eta_{it}$	Effizienz von Projektklasse 1. Ordnung $i$ in Periode $t$
$d_{ij}$	relativer effektiver Arbeitsaufwand von Projektklasse 2. Ordnung $j$ in Projektklasse 1. Ordnung $i$
$\gamma_{jt}$	Effizienz von Projektklasse 2. Ordnung $j$ in Periode $t$
$c_{jr}$	rel. effektiver Arbeitsaufwand von Kompetenz $r$ in Projektklasse 2. Ordnung $j$
$\theta_{rt}$	Effizienz von Kompetenz $r$ in Periode $t$

Nun der Reihe nach von hinten nach vorne: Für jede der  $R = 3$  Kompetenzen beobachten wir folgenden Kompetenzscorezuwachs:

$$z_{rt+1} = z_{rt} - \beta_r + \alpha_r \sum_{j=1}^m \frac{c_{jr}}{\theta_{rt}} \gamma_{jt} \xi_{jt} \quad (150)$$

Es gibt dabei  $m = 3$  Projektklassen 2. Ordnung und es gilt:

$$\theta_{rt} = \varphi(z_{rt}) \quad (151)$$

$$\gamma_{jt} = \left( \sum_{r=1}^R \frac{c_{jr}}{\theta_{rt}} \right)^{-1} \quad (152)$$

Die Variable  $\xi_{jt}$  misst, wie viel Zeit wir in Projektklasse 2. Ordnung  $j$  in Periode  $t$  stecken müssen, damit die Annahme über die gleichmäßige Projektabwicklung (vgl. Abschnitt 2.6 ab Seite 26) erfüllt ist. Diese Zeit ergibt sich aber einfach wie folgt:

$$\xi_{jt} := \sum_{i=1}^n \frac{d_{ij}}{\gamma_{jt}} \eta_{it} x_{it} \quad (153)$$

Dabei ist  $n = 3$  die Anzahl an Projektklassen 1. Ordnung.

Schließlich: Sei  $x_{it}$  jene Zeit, die wir in Projektklasse 1. Ordnung  $i$  stecken. Dann:

$$\eta_{it} = \left( \sum_{j=1}^m \frac{d_{ij}}{\gamma_{jt}} \right)^{-1} \quad (154)$$

In einer Formel:

$$\begin{aligned} z_{rt+1} &= z_{rt} - \beta_r + \alpha_r \sum_{j=1}^m \frac{c_{jr}}{\theta_{rt}} \gamma_{jt} \sum_{i=1}^n \frac{d_{ij}}{\gamma_{jt}} \eta_{it} x_{it} \\ &= z_{rt} - \beta_r + \alpha_r \sum_{i=1}^n \frac{\sum_{j=1}^m c_{jr} d_{ij}}{\theta_{rt}} \eta_{it} x_{it} \\ &= z_{rt} - \beta_r + \alpha_r \sum_{i=1}^n \frac{C_{ir}}{\theta_{rt}} \eta_{it} x_{it} \end{aligned} \quad (155)$$

Wir sehen, dass wir also die Projektklassen 2. Ordnung eliminieren können und wieder unser ursprüngliches Modell mit neuen Kompetenzen aber bewährten Projektklassen erhalten. Seien also die Matrizen  $d$  und  $c$  gegeben, dann lässt sich die  $n \times R$  Matrix  $C$  einfach mittels Matrixmultiplikation bestimmen:

$$C = d \cdot c \quad (156)$$

Die Werte für  $\alpha_r$  und  $\beta_r$  beziehen sich nun auf die neuen Kompetenzen und müssen neu geschätzt werden.

In Beispiel 68 von oben könnten etwa folgende (für  $c$  völlig fiktive) relativen effektiven Arbeitsaufwände zur Anwendung kommen:

$$d = \begin{bmatrix} 0.45 & 0.35 & 0.2 \\ 0 & 0.8 & 0.2 \\ 0 & 0 & 1 \end{bmatrix} \quad c = \begin{bmatrix} 0.5 & 0.4 & 0.1 \\ 0.5 & 0 & 0.5 \\ 0.75 & 0 & 0.25 \end{bmatrix} \quad C = d \cdot c = \begin{bmatrix} 0.55 & 0.18 & 0.27 \\ 0.55 & 0 & 0.45 \\ 0.75 & 0 & 0.25 \end{bmatrix}$$

Wir sehen in der Matrix  $c$ , dass Ausdauer (1. Spalte) also in allen drei Disziplinen notwendig ist. Die Beinkraft (2. Spalte) ebenso, allerdings nehmen wir an, dass sie beim Radfahren eine größere Bedeutung spielt als beim Laufen. Die Armkraft (3. Spalte) wird lediglich beim Schwimmen benötigt. Wir sehen, dass jetzt Radfahren und Laufen immer korreliert sind, auch wenn wir nur die eine oder andere Disziplin ausüben.

Dieser Umstand spiegelt sich auch in der Matrix  $C$  wider – Duathlon und Laufen hängen (dank der neuen Kompetenzen Ausdauer und Beinkraft) viel stärker zusammen als zuvor.

Diese Vorgehensweise lässt sich beliebig oft anwenden. Wir könnten also (wenn wir das für sinnvoll hielten) 500 Zwischenschichten einführen, da wir das komplizierte Modell leicht wieder zu unserem einfachen Basismodell zusammenpressen können.

## 7.5. Gezieltes Training von Kompetenzen

Den Aspekt des Trainings, also die gezielte Effizienzverbesserung von Kompetenzen gegen Entgeltzahlung, haben wir komplett ausgelassen. Trotz der negativen Erträge ist es durchaus möglich, dass die Zielfunktion unterm Strich steigt, wenn wir einzelne Kompetenzen gezielt fördern. Die Chancen dafür steigen, je langfristiger wir denken.

Eine stark abgeschwächte Form des Trainings haben wir im Triathlonbeispiel gesehen. Dort war es möglich, die Laufleistung gezielt zu verbessern, da Laufen nicht nur als Kompetenz, sondern auch als Projektklasse zur Verfügung gestanden ist. So konnte in den Beispielen die Effizienz des Duathlons durch Hineinmischen von Laufen stark verbessert werden. Allerdings haben wir hierfür nicht zahlen müssen, im Gegenteil: Wir haben sogar etwas erhalten.



## A. Zusammenfassung auf Deutsch

Diese Arbeit befasst sich mit Projektportfoliooptimierung unter einem Kompetenzentwicklungsmodell. Sie basiert auf [Gut11], dort wird ein Spezialfall des zu Grunde liegenden Modells ausführlich diskutiert und der allgemeine Fall aufgeschrieben. Wir nehmen in dieser Arbeit den allgemeinen Fall genauer unter die Lupe.

Uns stehen Projekte aus diversen Projektklassen zur Verfügung. Für die Erledigung der Projekte aus den Projektklassen sind verschiedene Kompetenzen in unterschiedlichem Ausmaß erforderlich. Machen wir von einer Kompetenz oft Gebrauch, werden wir besser in ihr und können in weiterer Folge Projekte, welche diese Kompetenzen benötigen, schneller bearbeiten. Das gleiche gilt umgekehrt.

Ziel ist es, die verfügbare Arbeitszeit derart auf diese Projektklassen aufzuteilen, dass der Gesamtertrag maximal ist. Dabei unterscheiden wir zwischen der kurzfristigen und der langfristigen Sichtweise, welche in unmittelbarem Gegensatz zueinander stehen. Langfristig gute Lösungen sind in der Regel kurzfristig schlecht und kurzfristig gute Lösungen sind meist langfristig unbefriedigend.

Zahlreiche mathematische Resultate unterstützen uns dabei Algorithmen bereitzustellen, welche uns bei der Wahl des „richtigen“ Projektportfolios unterstützen, indem sie nach lokal optimalen Lösungen suchen. Dabei untersuchen wir zunächst die kurzfristige und langfristige Sichtweise, bevor wir uns einem mittelfristigen Kompromiss zuwenden.

In **Kapitel 2** erläutern wir die einzelnen Bestandteile des Basismodells, und fassen das Problem zusammen. Abschließend stellen wir fest, dass das Modell überparametrisiert ist und zeigen, wie wir einige Parameter normieren könnten.

Die Generierung gleichverteilter Zufallslösungen mit Hilfe der Inversionsmethode ist Thema von **Kapitel 3**. Wir überlegen uns, wie die Dichtefunktionen aussehen müssen, um tatsächlich eine Gleichverteilung auf der zulässigen Menge zu erhalten.

Am Ende von **Kapitel 4** finden wir einen lokalen Optimierungsalgorithmus für die kurzfristige Sichtweise vor, der auf Gradientenanstieg beruht. Allerdings ist der gewöhnliche Gradient eine unzulässige Richtung, folglich stellen wir eine Möglichkeit bereit, diesen zu korrigieren. Um notwendige Bedingungen für lokale Optima zu erhalten, adaptieren wir die Karush-Kuhn-Tucker-Bedingungen auf unsere Problemstellung. Weiters überlegen wir uns einen Weg, Sattelpunkten entfliehen zu können.

Wir beweisen in **Kapitel 5**, dass es Fixpunkte (Gleichgewichtszustände) für die Effizienzwerte des Kompetenzentwicklungssystems gibt, wenn wir in jeder Periode unsere verfügbare Arbeitszeit immer gleich auf die Projektklassen aufteilen. Und wir zeigen, dass der Fixpunkt – von einem Sonderfall abgesehen – eindeutig ist.

Wir müssen zur Kenntnis nehmen, dass idR keine geschlossene Formel oder analytische Rechenvorschrift existiert um das Gleichgewicht zu berechnen. Daher entwickeln wir – inspiriert durch die vorangehenden Resultate – die „Holzhackermethode“, ein Iterati-

onsverfahren, das uns bewiesenermaßen für ein gegebenes Portfolio einen Fixpunkt des Kompetenzentwicklungssystems in endlicher Laufzeit zurückgibt.

Wir entwickeln ein Iterated Local Search Verfahren, mit dem wir eine Liste von Projektportfolios mit langfristigem Optimalitätspotenzial erstellen. Dabei gehen wir von zufällig generierten Projektportfolios und den dazupassenden Erträgen im Gleichgewicht aus und verbessern die Lösung mit Hilfe des Pattern Search Algorithmus. Leider stellen wir fest, dass es Ebenen geben kann, also zusammenhängende Bereiche von Portfolios, welche den gleichen Ertrag aufweisen. Im Anhang skizzieren wir, wie wir mit Hilfe der Clusteranalyse die Fülle an ähnlichen und gleich guten Lösungen komprimieren können.

In **Kapitel 6** diskutieren wir anhand von einigen Beispielen den sich ergebenden Trade-Off zwischen der kurzfristigen und langfristigen Sichtweise und schreiben einen Rolling Horizon Algorithmus auf, der nach einem Kompromiss zwischen beiden Sichtweisen sucht.

Rolling Horizon eignet sich hervorragend dazu Lösungen mit langfristigem Optimalitätspotenzial auf die kurze Sichtweise hin zu adaptieren. Je größer der Horizontparameter und je kleiner der Zinssatz, desto mehr wiegt die Zukunft und desto eher werden langfristig gute Lösungen generiert.

Die Wahl des Horizontparameters erfordert jedoch Fingerspitzengefühl und Erfahrung. Ist dieser zu klein eingestellt, so vergeben wir die Chance auf langfristigen Erfolg. Ist dieser umgekehrt zu groß eingestellt, so dauert es zu lange, bis sich die anfänglichen Investitionen in die Zukunft amortisieren.

In **Kapitel 7** zeigen wir eine Möglichkeit, wie wir mit den oben angesprochenen Ebenen umgehen können. Weiters konstatieren wir, dass es in der Praxis Preis- und Zinsschwankungen gibt und zeigen, dass wir das nunmehrige stochastische Optimierungsproblem mit den Techniken der vorherigen Kapitel lösen können, wenn auch sicher nicht auf die effizienteste Art und Weise.

## B. Summary in English

This master thesis deals with project portfolio optimization under a competence development model. It is based on [Gut11], where a special case of the underlying model is taken into account and the general case is mentioned. Here we want to get deeper insights into this general case.

Projects from a variety of project classes are available. Several competencies to a different extent are needed to finish the projects of the project classes. The more we make use of a certain competence the better we get and we can use it in a more efficient manner in the future. Therefore we can finish projects that require this competence faster.

Now our goal is to assign our available working time to the project classes such that the overall return is maximized. Thereby we distinguish between the short-term and the long-term view which are in conflict to each other. Solutions with marvellous long-term returns generally suffer from poor short-term returns. And solutions with great short-term returns usually show a bad performance in the long run.

Numerous mathematical results support us to find algorithms that support us to find the “right” project portfolio by searching for local optima. At first we study the short-term view, then we turn to the long-term view. In the end we try to find a medium-term compromise.

In **section 2** we explain the components of the basic model and summarize the optimization problem. We also discover that our model is overparametrized and show that some parameters could be standardized.

In **section 3** we use the inverse transform sampling technique to generate uniformly distributed random solutions on the feasible set. Therefore we derive density functions that fulfil the requirements.

At the end of **section 4** we find a local optimization algorithm for the short-term view that is based on gradient ascent. However, as we see the gradient is an infeasible direction, we offer a suitable possibility to adjust the gradient. From the Karush-Kuhn-Tucker conditions we get proper necessary conditions for local optima in our model. Unfortunately these conditions are not sufficient, hence, we think about a way how to escape saddle points.

We show the existence of fixed points (equilibrium states) for the efficiencies of the competence development system in **section 5** if we apply the same policy in each period. We show that the fixed point is unique except for one special case.

Generally no closed formula or analytical calculation rule exist to compute the equilibrium. Inspired by the preceding results we evolve the “Holzhackermethode” (lumberjack method) – an iterative method that returns a fixed point of the competence development system for a given constant project portfolio in finite running time.

We develop an iterated local search algorithm that establishes a list of solutions with the potential of being locally optimal in the long run. Therefore we start with randomly generated project portfolios and their returns in the equilibril state and improve them by using pattern search. Regrettably there can exist planes – i.e. connected areas of portfolios with the same return. In the appendix we delineate how we can compress the set of similar solutions by using cluster analysis.

On the basis of some examples we, in **section 6**, discuss the trade-off between the short-term view and the long-term view and note down a rolling horizon algorithm that seeks a compromise between both views.

Rolling horizon is well appropriate to adapt solutions with long-term optimality potential to the short-term view. The higher the horizon parameter and the smaller the interest rate, the more important the future and the higher the opportunity to obtain good solutions in the long run.

The choice of the horizon parameter requires a high level of intuition and experience. If it is too small, then we lose the chance to receive solutions with a good long-term return. If, on the other hand, the parameter is too high, then it might last too long until the initial investments amortize.

In **section 7** we discuss how to handle with the above mentioned planes. Furthermore we state that prices and interest rates fluctuate in practice and we show that the corresponding stochastic optimization problem can be solved by the techniques of the previous sections, although they are for sure not the most efficient ones.

## C. Verwendete Programme und Anmerkungen zur Umsetzung

Diese Arbeit wurde mit  $\LaTeX$  erstellt. Dabei habe ich mich des Programms TeXnic-Center sowie der  $\LaTeX$ -Distibution MikTeX bedient. Aufwändigere Simulationen (zB Berechnung der lokalen Optima) habe mit Hilfe der Programmiersprache Java und der Benutzeroberfläche NetBeans IDE durchgeführt. Die Bilder zu Beginn des Kapitels 2 habe ich mit Microsoft Paint, sowie Adobe Photoshop gezeichnet. Kleinere Simulationen, die Auswertungen der Ergebnisse und die Grafiken habe ich mit R erstellt.

Der gesamte Programmcode, welcher für diese Arbeit erforderlich war, umfasst unzählige Codezeilen. Auf jeden Fall so viele, dass ich davon abgesehen habe, diesen hier zu veröffentlichen. Stattdessen habe ich mich bemüht, die Algorithmen möglichst so aufzuschreiben, dass sie nahezu 1 zu 1 umgesetzt werden können.

Einen Gutteil der Codezeilen musste ich für die Behebung von numerischen Fehlern und Ungenauigkeiten aufwenden. Dazu zählen Rundungsfehler, welche bei der lokalen Optimierung aufgetreten sind. So sind nach dem Linesearch oftmals Werte für  $x_{it}$  in der Größenordnung von  $\pm 10^{-17}$  entstanden. Solche Werte habe ich dann manuell auf 0 gesetzt, um bei der Bestimmung der Menge  $I_t(x)$ , für welche  $\pm 10^{-17} \neq 0$  gelten würde, keine bösen Überraschungen zu erleben.

Im Laufe der lokalen Optimierung sind numerisch bedingte Rundungsfehler aufgetreten, sodass technisch gesehen die Bedingung

$$\sum_{i=1}^n x_{it} = \sigma_t$$

nicht erfüllt war, bzw. sich der Fehler fortgepflanzt hat. Um diese Fehler in Grenzen zu halten, habe ich regelmäßig wie folgt korrigierend eingegriffen:

$$\begin{aligned} \theta_t &\leftarrow \sum_{i=1}^n x_{it} \quad \forall t \\ x_{it} &\leftarrow \frac{x_{it}}{\theta_t} \cdot \sigma_t \quad \forall t \end{aligned}$$



## D. Verzeichnisse

### D.1. Abbildungsverzeichnis

1.	Grenzwinne im Triathlonbeispiel in der üblichen Konstellation . . . . .	19
2.	Darstellung des Projekts Triathlon-Kurzdistanz . . . . .	23
3.	Veranschaulichung der Lernfunktion $\varphi(\cdot)$ für verschiedene $a$ und $b$ . . . . .	25
4.	Unterschiedliche Projektabwicklungspolitiken . . . . .	27
5.	Transformation der Lernfunktionen in Beispiel 11 . . . . .	33
6.	1000 gleichverteilte Zufallslösungen für $n = 3$ und $T = 1$ . . . . .	38
7.	Entwicklung des korrigierten Gradienten im 1. Gradientenbeispiel . . . . .	46
8.	Entwicklung des korrigierten Gradienten im 2. Gradientenbeispiel . . . . .	47
9.	Darstellung der Zielfunktion und der lokalen Optimierung im lokalen Optimierungsbispiel . . . . .	55
10.	Darstellung der Inselbetriebe in Beispiel 30 . . . . .	62
11.	Veranschaulichung von Eindeutigkeitssatz A . . . . .	67
12.	Veranschaulichung von Eindeutigkeitssatz C . . . . .	67
13.	Veranschaulichung von Eindeutigkeitssatz AC ( $\bar{\gamma}_{r_{min}} < 1$ ) . . . . .	69
14.	Veranschaulichung von Eindeutigkeitssatz AC ( $\bar{\gamma}_{r_{min}} = 1$ ) . . . . .	69
15.	Veranschaulichung von Eindeutigkeitslemma ABC ( $\bar{\gamma}_r \geq \tilde{\gamma}_r \forall r \in \bar{\mathcal{R}}_B$ ) . . . . .	71
16.	Veranschaulichung von Eindeutigkeitslemma ABC ( $\exists r \in \bar{\mathcal{R}}_B : \bar{\gamma}_r < \tilde{\gamma}_r$ ) . . . . .	72
17.	Entwicklung von $d_{irt}$ und $\gamma_{rt}$ im Oszillationsbeispiel . . . . .	76
18.	Entwicklung der Effizienzwerte im Stabilitätsbeispiel . . . . .	82
19.	Veranschaulichung der Holzhackermethode . . . . .	88
20.	Schematische Darstellung des Grenzwinns im Triathlonbeispiel . . . . .	89
21.	Grenzwinne und ausgewählte Lösungen im Triathlonbeispiel in der üblichen Konstellation . . . . .	90
22.	Veranschaulichung des Pattern Search Algorithmus . . . . .	92
23.	Grenzwinne im 1. Ebenenbeispiel . . . . .	96
24.	Grenzwinne im 2. Ebenenbeispiel . . . . .	97
25.	Grenzwinne und lokale Grenzoptima im 3. Ebenenbeispiel . . . . .	98
26.	Attraktionsbecken im Triathlonbeispiel . . . . .	102
27.	Vergleich dreier Lösungsvarianten in Beispiel „Radtop“ . . . . .	105
28.	Vergleich dreier Lösungsvarianten in Beispiel „Schwimmtop“ . . . . .	107
29.	Vergleich dreier Lösungsvarianten in Beispiel „Perfektion“ . . . . .	109
30.	Schematische Darstellung des Rolling Horizon Verfahrens . . . . .	110
31.	Vergleich zweier Lösungsvarianten in Beispiel „Perfektion“ . . . . .	115
32.	Vergleich zweier Lösungsvarianten in Beispiel „Radtop“ . . . . .	115
33.	k-Schablonen im Triathlonbeispiel mit $k = 12$ und variablem $\delta$ . . . . .	116
34.	k-Schablonen im Triathlonbeispiel mit variablem $k$ , $\delta = 1$ und $\delta = 1.1$ . . . . .	117
35.	Darstellung der Clusteranalyse im 2. Ebenenbeispiel . . . . .	120
36.	Darstellung der Clusteranalyse im 3. Ebenenbeispiel . . . . .	121

## D.2. Tabellenverzeichnis

1.	Das Projekt Triathlon-Kurzdistanz . . . . .	23
2.	Das Projekt Triathlon-Sprint . . . . .	23
3.	Das Projekt Triathlon-Kurzdistanz revisited . . . . .	26
4.	Darstellung der 5 besonderen Punkte im lokalen Optimierungsbeispiel . . . . .	56
5.	Die Projektklasse Triathlon: Daten für das Szenario im Oszillationsbeispiel . . . . .	76
6.	Die Projektklasse Triathlon: Szenarien für das Stabilitätsbeispiel . . . . .	78
7.	Gleichgewichtszustände im Stabilitätsbeispiel . . . . .	81
8.	Übersicht über alle ausgewählten Punkte im Triathlonbeispiel . . . . .	91
9.	Einsparungspotenzial bei den Richtungen in Pattern Search . . . . .	93
10.	Übersicht über alle lokalen Grenzoptima im Triathlonbeispiel . . . . .	101
11.	Gewicht der Zukunft für ausgewählte $k$ und $\delta$ . . . . .	113
12.	12-Schablonen des Triathlonbeispiels „Radtop“ . . . . .	116

### D.3. Algorithmenverzeichnis

1.	Generiere gleichverteilte Zufallslösung . . . . .	38
2.	Berechne den Gradienten . . . . .	42
3.	Berechne den korrigierten Gradienten . . . . .	45
4.	Berechne kurzfristiges lokales Optimum . . . . .	54
5.	Bestimme alle minimalen Inselbetriebe . . . . .	61
6.	Berechne Gleichgewichtszustand – Holzhackermethode . . . . .	85
7.	Berechne einen Kandidaten für lokales Grenzoikum – Pattern Search .	95
8.	Prüfe auf Ebene . . . . .	99
9.	Generiere Kandidatenliste für lokale Grenzoikima – Pattern Search ILS .	100
10.	Berechne mittelfristiges lokales Optimum – Rolling Horizon . . . . .	112

## D.4. Literatur

- [AC07] ALBA, E. ; CHICANO, F.: Software project management with GAs. In: *Information Science* 177 (2007), S. 2380–2401
- [ADLL02] ATTIA, E.-A. ; DUQUENNE, P. ; LE-LANN, J.-M.: *Decision-based genetic algorithms for solving multi-period project scheduling with dynamically experienced workforce*. 9th International Conference on Modeling, Optimization and Simulation (MOSIM'12), 6.-8. Juni 2012, Bordeaux, Frankreich, 2002
- [AE90] ARGOTE, L. ; EPPLE, D.: Learning curves in manufacturing. In: *Science* 23 (1990), S. 920–924
- [AGM07] ARMBRUSTER, D. ; GEL, E. S. ; MURAKAMI, J.: Bucket brigades with worker learning. In: *European Journal of Operational Research* 176 (2007), S. 264–274
- [Bas00] BASSETT, M.: Assigning projects to optimize the utilization of employees' time and expertise. In: *Computers and Chemical Engineering* 24 (2000), S. 1013–1021
- [BL97] BIRGE, John R. ; LOUVEAUX, François: *Introduction to Stochastic Programming*. Springer, 1997 (Springer Series in Operation Research)
- [Bom98] BOMZE, Immanuel: *Mathematik für Studierende der Volkswirtschaft und Statistik - Skriptum zur Vorlesung von Prof. Dr. Immanuel Bomze im Studienjahr 2005/2006*. Universitätsstraße 5/9, 1010 Wien : Institut für Statistik und Decision Support Systems, 1998
- [CE05] CHEN, A. N. K. ; EDGINGTON, T. M.: Assessing value in organizational knowledge creation: considerations for knowledge workers. In: *MIS Quarterly* 29 (2005), S. 279–309
- [Chr03] CHRISTENSEN, Clayton M.: *The innovator's dilemma: the revolutionary book that will change the way you do business*. New York : HarperCollins, 2003
- [DAE95] DARR, E. D. ; ARGOTE, L. ; EPPLE, D.: The acquisition, transfer, and depreciation of knowledge in service organizations: productivity in franchises. In: *Management Science* 41 (1995), S. 1750–1761
- [EM08] EISELT, H. A. ; MARIANOV, V.: Employee positioning and workload allocation. In: *Computers & Operations Research* 35 (2008), S. 513–524
- [FAK11] FIKRI, M. ; ALAOUI, A. E. H. ; KHOMSSI, M. E.: Assignment staff with dynamic competencies in multi-projects and multi-periods: modelling and solving by a hybridization of ant colony optimization. In: *International Journal of Computer Science and Network Security* 140 (2011), S. 227–240

- [FH07] FREUND, Roland W. ; HOPPE, Ronald H.: *Stoer/Bulirsch: Numerische Mathematik 1*. 10., neu bearbeitete Auflage. Springer, 2007
- [Fio07] FIORETTI, G.: The organizational learning curve. In: *European Journal of Operational Research* 177 (2007), S. 1375–1384
- [GKR07] GUTJAHR, W. J. ; KATZENSTEINER, S. ; REITER, P.: A VNS algorithm for noisy problems and its application to project portfolio analysis. In: *J. Hromkovic et al (eds) Proceedings of the SAGA 2007 (stochastic algorithms: foundations and applications)* Bd. 4665. Springer Lecture Notes in Computer Science, 2007, S. 93–104
- [GKR<sup>+</sup>08a] GUTJAHR, W. J. ; KATZENSTEINER, S. ; REITER, P. ; STUMMER, C. ; DENK, M.: Competence-driven project portfolio selection, scheduling and staff assignment. In: *Central European Journal of Operations Research* 16 (2008), S. 281–306
- [GKR<sup>+</sup>08b] GUTJAHR, W. J. ; KATZENSTEINER, S. ; REITER, P. ; STUMMER, C. ; DENK, M.: Multi-objective decision analysis for competence-oriented project portfolio selection / Department of Statistics and Decision Support Systems. University of Vienna, 2008. – Forschungsbericht
- [GR10] GUTJAHR, W. J. ; REITER, P.: Bi-objective project portfolio selection and staff assignment under uncertainty. In: *Optimization Methods and Software* 59 (2010), S. 417–445
- [Gut11] GUTJAHR, Walter J.: Optimal dynamic portfolio selection for projects under a competence development model. In: *OR Spectrum* 33 (2011), Januar, S. 173–206
- [HK08a] HEIMERL, C. ; KOLISCH, R.: Scheduling and staffing multiple projects with a multi-skilled workforce / TUM Business School. Technical University of Munich, Germany, 2008. – Forschungsbericht
- [HK08b] HEIMERL, C. ; KOLISCH, R.: Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets / TUM Business School. Technical University of Munich, Germany, 2008. – Forschungsbericht
- [Kol08] KOLONKO, Michael: *Stochastische Simulation*. Vieweg+Teubner, 2008
- [LK02] LOCH, Ch. H. ; KAVADIAS, S.: Dynamic portfolio selection of NPD programs using marginal returns. In: *Management Science* 48 (2002), S. 1227–1241
- [LS12] LIESIÖ, J. ; SALO, A.: Scenario-based portfolio selection of investment projects with incomplete probability and utility information. In: *European Journal of Operational Research* 217 (2012), S. 162–172

- [Möh06] MÖHRING, Rolf H.: *Algorithmische Diskrete Mathematik III, Angewandte Netzwerkoptimierung, Kapitel 2: Grundlagen der nichtlinearen (konvexen) Optimierung. Lecture Notes.* <http://page.math.tu-berlin.de/~moehring/adm3/adm3-2006/Kapitel/adm3-mitschrift-par2.pdf>, SS 2006. – Link aktualisiert am 01.08.2013
- [NGM07] NGWENYAMA, O. ; GUERGACHI, A. ; McLAREN, T.: Using the learning curve to maximize IT productivity: A decision analysis model for timing software upgrades. In: *International Journal of Production Economics* 105 (2007), S. 524–535
- [PS07] PENDHARKAR, P. C. ; SUBRAMANIAN, G. H.: An empirical study of ICASE learning curves and probability bounds for software development effort. In: *European Journal of Operational Research* 183 (2007), S. 1086–1096
- [Pöt08] PÖTSCHER, Benedikt M.: *Asymptotische Statistik / Ökonometrie.* Lecture Notes, SS 2008
- [Seb04] SEBER, George A. F.: *Multivariate Observations.* Wiley, 2004 (Wiley series in probability and statistics)
- [SKG09] STUMMER, C. ; KIESLING, E. ; GUTJAHR, W. J.: A multicriteria decision support system for competence-driven project portfolio selection. In: *International Journal of Information Technology & Decision Making* 8 (2009), S. 379–401
- [SY06] SUN, Wenyu ; YUAN, Ya-Xiang: *Optimization Theory and Methods.* Springer, 2006 (Springer Series in Optimization and its Applications). – Kapitel 2 über Linesearch downloadbar: [http://www.springer.com/cda/content/document/cda\\_downloaddocument/9780387249759-c2.pdf](http://www.springer.com/cda/content/document/cda_downloaddocument/9780387249759-c2.pdf) – Link aktualisiert am 01.08.2013
- [TYD06] TECHAWIBOONWONG, A. ; YENRADEE, P. ; DAS, S. K.: A master scheduling model with skilled and unskilled temporary workers. In: *International Journal of Production Economics* 103 (2006), S. 798–809
- [WS06] WU, M.-C. ; SUN, S.-H.: A project scheduling and staff assignment model considering learning effect. In: *Journal of Advanced Manufacturing Technology* 28 (2006), S. 1190–1195
- [YFIN06] YOSHIMURA, M. ; FUJIMI, Y. ; IZUI, K. ; NISHIWAKI, S.: Decision-making support system for human resource allocation in product development projects. In: *International Journal of Production Research* 44 (2006), S. 831–848

Informationen zum Triathlon unter <http://de.wikipedia.org/wiki/Triathlon> (Link aktualisiert am 01.08.2013)

## E. Lebenslauf

### Zu meiner Ausbildung

2005	Matura an der Bundeshandelsakademie (BHAK) business.academy.donaustadt in 1220 Wien
2005 – 2009	Bakkalaureatsstudium Statistik an der Universität Wien
2009 – 2014	Magisterstudium Statistik an der Universität Wien
2010 – 2011	Zivildienst beim Fonds Soziales Wien im Tageszentrum für Seniorinnen und Senioren Donauefeld

### Zu meinen Abschlussarbeiten

Bakkalaureatsarbeit 1	Clustern mit Affinity Propagation <i>Betreuer:</i> Univ.-Prof. Dr. Marcus Hudec
Bakkalaureatsarbeit 2	Iterated Local Search und Ant Colony Optimization für das Travelling Salesman Problem <i>Betreuer:</i> Univ.-Prof. Dr. Walter J. Gutjahr
Magisterarbeit	Projektportfoliooptimierung unter einem Kompetenzentwick- lungsmodell – Kurz- und langfristige Lösungstechniken, ein mittelfristiger Kompromiss und weitere interessante Einblicke <i>Betreuer:</i> Univ.-Prof. Dr. Walter J. Gutjahr

### Zu meiner universitären Praxis

2007 – 2009	Tutor für die LV Analysis und Praktische Mathematik von Univ.-Prof. Dr. Werner Schachinger an der Universität Wien (2 Semester)
2009 – 2014	Tutor für die LV Statistisches Programmieren von Univ.-Prof. Dr. Marcus Hudec und Dr. Andreas Baierl an der Universität Wien (6 Semester)
2011	Tutor für die LV Computational Statistics von Univ.-Prof. Dr. Marcus Hudec und Dr. Andreas Baierl an der Universität Wien (1 Semester)

### Zu meinen sonstigen Funktionen

seit 2011	Jobreferent der Österreichischen Statistischen Gesellschaft
seit 2011	Ballreferent des Badmintonvereins Sportunion BC PRA
seit 2012	Mannschaftsführer der 3. Mannschaft des Badmintonvereins Sportunion BC PRA
seit 2013	Finanzreferent des Badmintonvereins Sportunion BC PRA