



Forschungsreihe des Fachbereichs
Ingenieurwissenschaften und Mathematik

Angewandte mathematische Modellierung und Optimierung

Ausgewählte Modelle, Methoden, Fallstudien

Hermann-Josef Kruse, Timo Lask (Hrsg.)



Angewandte Mathematische Modellierung & Optimierung

Band 4



FH Bielefeld
University of
Applied Sciences

Impressum

**Forschungsreihe des Fachbereichs
Ingenieurwissenschaften und Mathematik**

ISSN

2196-6192

Erscheinungsort

Bielefeld

Verantwortliche und ausführende Stelle (Herausgeber)

im Sinne des § 55 I RStV und des Presserechts NRW:
Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und Mathematik
Interaktion 1
33619 Bielefeld

Der Fachbereich wird innerhalb der Fachhochschule
Bielefeld vom Dekan Prof. Dr.-Ing. Prof. h.c. Lothar Budde
geleitet und vertreten.

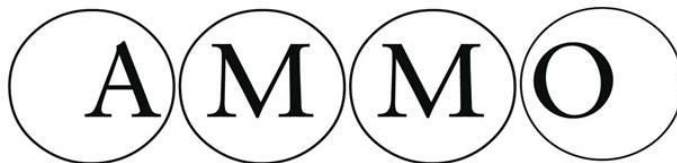
Verantwortliche Chefredaktion

Prof. Dr.-Ing. Rolf Naumann
Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und Mathematik
Interaktion 1
33619 Bielefeld
Telefon +49.521.106-7483
Telefax +49.521.106-7191
rolf.naumann@fh-bielefeld.de

Angewandte mathematische Modellierung und Optimierung

Ausgewählte Modelle, Methoden, Fallstudien

Hermann-Josef Kruse, Timo Lask (Hrsg.)



Angewandte Mathematische Modellierung & Optimierung

Vorwort

Im Jahre 2008 gründeten fünf Wissenschaftlerinnen und Wissenschaftler, die alle im Bachelor-Studiengang *Angewandte Mathematik* und auch im Master-Studiengang *Optimierung und Simulation* des Fachbereichs *Ingenieurwissenschaften und Mathematik* der Fachhochschule Bielefeld lehr(t)en, den Forschungs- und Entwicklungsschwerpunkt *Angewandte Mathematische Modellierung und Optimierung* (FSP AMMO). Mittlerweile ist die Gruppe auf zwölf Personen angewachsen.¹

Der Hauptzweck dieses Zusammenschlusses ist es, das spezifische Wissen und die Erfahrungen der einzelnen Wissenschaftlerinnen und Wissenschaftler, die ihre Forschungsausrichtungen auf unterschiedlichen Gebieten der Mathematik haben und weitreichende Kenntnisse aus verschiedenen Praxisumfeldern mitbringen, im gemeinsamen Schnittbereich der mathematischen Modellierung und Optimierung zu bündeln und aus diesem Kernbereich gemeinsame Forschungs- und Entwicklungsprojekte zu generieren. Seit der Gründung von AMMO bis heute sind viele unterschiedliche Projekte initiiert und durchgeführt worden.²

Zum siebenjährigen Bestehen des Forschungs- und Entwicklungsschwerpunktes wurde die Idee geboren, das facettenreiche Gebiet der angewandten mathematischen Modellierung und Optimierung durch typische Problemfelder und dazu passende Lösungsansätze anschaulich zu charakterisieren und in Form eines Buches zu präsentieren. Das Ziel dieses Buches ist es also, den systematischen Lösungsweg für ein Praxisproblem von der Problembeschreibung über die mathematische Modellierung und die Entwicklung bzw. Auswahl geeigneter Lösungsverfahren und deren Anpassung bis hin zur Lösungsinterpretation anhand von einigen Fallstudien nachvollziehbar aufzuzeigen. Die vorgestellten Beispiele stammen aus den Forschungsbereichen der involvierten AMMO-Mitglieder, teilweise in Zusammenarbeit mit anderen Wissenschaftlern und unterstützt durch Studierende des Bachelor-Studienganges *Angewandte Mathematik* und des Master-Studienganges *Optimierung und Simulation* im Rahmen von Qualifizierungsarbeiten.

Das Buch enthält sieben Beiträge, die sich sowohl in den Anwendungsgebieten der mathematischen Modellierung und Optimierung als auch in der Ausrichtung auf diese beiden Schwerpunkte unterscheiden und somit eine breite Palette der angewandten mathematischen Modellierung und Optimierung bieten. Die ausgewählten Anwendungsgebiete sind die Biologie (im weitesten Sinne), die Personalplanung, die Entsorgungswirtschaft, die Ener-

¹Näheres hierzu findet man im Nachwort dieses Buches ("Wer und Was ist AMMO?").

²Hierzu sei auf den Link <http://www.fh-bielefeld.de/ammo> verwiesen.

gieversorgungswirtschaft sowie das Gebiet der Strukturoptimierung. Im Buch wurde die Reihenfolge der Beiträge auch im Hinblick auf die genannten Schwerpunkte *Modellierung* und *Optimierung* vorgenommen. Zwar enthalten die Beiträge immer auch beide Aspekte, zumal die Modellbildung und die Verfahrensanwendung eine Symbiose bilden, allerdings in unterschiedlichem Ausmaß.

So bilden die beiden Beiträge *Mathematische Modellierung des Laubharkproblems als spezielle Ausprägung von allgemeinen Entsorgungsprozessen* und *Heuristische Lösungsstrategien für das Laubharkproblem* einen Brückenschlag zwischen diesen beiden Schwerpunkten. Während sich der erstgenannte Beitrag allein auf die Modellierung des herbstlichen Laubharkproblems konzentriert und aufzeigt, wie ein real existierendes Alltagsproblem in ein mathematisches Optimierungsmodell überführt werden kann, liefert der zweitgenannte einige effiziente Lösungsverfahren für dieses Optimierungsproblem und zeigt dabei, dass die Komplexität dieser recht simpel erscheinenden Alltagsproblematik dazu zwingt, auf heuristische Lösungsverfahren zurückzugreifen und demzufolge mit möglicherweise nur suboptimalen Lösungen zufrieden zu sein. Dabei dient das Laubharken als ein exemplarischer Entsorgungsprozess. Diese beiden Beiträge sind mittig eingereiht, um einen fließenden Übergang zwischen (vornehmlich) modellorientierten Beiträgen im ersten Teil des Buches zu den (vornehmlich) verfahrensorientierten Beiträgen im zweiten Teil zu bilden.

Die beiden ersten Beiträge befassen sich im weitesten Sinne mit Problemen aus der Biologie, genauer gesagt der Biostatistik bzw. statistischen Ökologie einerseits und der Biotechnologie andererseits. Der Beitrag *Schätzung eines Libellenbestands im Naturschutzgebiet Heiliges Meer mit der Rückfangmethode der Biostatistik* behandelt eine Modellierungsaufgabe aus dem Gebiet der statistischen Ökologie. Es kommen verschiedene auf der so genannten Rückfangmethode basierende Verfahren zur Schätzung einer Populationsgröße zum Einsatz. Aus mathematischer Sicht besteht das Ziel hauptsächlich darin, grundlegende Aspekte dieser Schätzverfahren auf exemplarische Weise derart darzustellen, dass sie auch auf ähnlich gelagerte statistische Fragestellungen angewendet und angemessen interpretiert werden können.

Im zweiten Beitrag *Optimierung von Produktionsprozessen in der Biotechnologie* wird eine allgemeine Vorgehensweise vorgestellt, wie sich bestimmte Produktionsprozesse in der Biotechnologie mit dem bikriteriellen Ziel der Steigerung der Produktausbeute und der Verringerung der Prozesskosten als mathematisches Optimierungsproblem formulieren lassen. Dazu wird ein generelles Konzept zur Etablierung einer automatisierten Steuerung bzw. Regelung des Bioprozesses aufgezeigt. Diese Konzeptentwicklung wird anhand zweier Beispiele,

der Xanthanproduktion des Bakteriums *Xanthomonas campestris* und der Antikörperproduktion der CHO-Zellen, konkretisiert.

Auch im dritten Beitrag *Simulation und Optimierung der Personalplanung im Pflegebereich von Krankenhäusern durch Petri-Netz-Modelle* steht die mathematische Modellbildung im Fokus, allerdings zu einer Problematik aus dem wirtschaftswissenschaftlichen Teilgebiet der Personalplanung und zweckdienlich auf Simulationsmodelle ausgerichtet. Hierbei geht es konkret um die Personalplanung von Pflegekräften in Krankenhäusern, wobei verschiedene Arbeitszeitkonzepte zur Diskussion stehen. Mit Hilfe von Simulationsmodellen auf der Basis von Petri-Netzen werden diese Konzepte vor ihrer konkreten Umsetzung auf Machbarkeit hinsichtlich relevanter Kriterien und Vorgaben überprüft, wodurch ein wichtiger Beitrag zur Entscheidungsunterstützung bei der Wahl eines geeigneten Arbeitszeitmodells geliefert wird.

Die beiden letzten Beiträge sind in englischer Sprache verfasst. Im Mittelpunkt des Beitrages *Biomimetic Methods for Power Plant Optimization* steht die Methodik zur Lösung eines konkreten Optimierungsproblems aus dem Gebiet der Energieversorgung. Es wird gezeigt, dass auf die üblicherweise vorgenommene Beschreibung von Kraftwerksprozessen durch angepasste gemischt-ganzzahlige lineare Optimierungsmodelle weitgehend verzichtet werden kann, indem Lösungsansätze aus der Bionik zum Einsatz kommen. Hierbei wird explizit auf die beiden bionischen Verfahren Simulated Annealing und Artificial Bee Colony Algorithm zurückgegriffen. Last but not least beschäftigt sich der Beitrag *Applications of Shape and Topology Optimization* mit der Strukturoptimierung beim computergestützten Design von innovativen Werkstoffen und Hochleistungs-Elektromotoren. Die Konstruktion dieser neuen Materialsysteme wird durch die Optimierung ihrer Betriebseigenschaften bestimmt. Das Augenmerk ist dabei auf schnelle und robuste numerische Lösungsverfahren für die restringierten Optimierungsprobleme ausgerichtet, wobei die Nebenbedingungen sich im Wesentlichen als partielle Differentialgleichungen (PDEs) stellen. Zur Lösung werden primal-duale Innere-Punkte-Verfahren herangezogen.

Das Buch schließt mit einem Nachwort (*Wer und Was ist AMMO?*), welches eine kurze Darstellung des Forschungs- und Entwicklungsschwerpunktes AMMO im Verlaufe des siebenjährigen Bestehens von AMMO umfasst. Zu guter Letzt werden die an diesem Buch beteiligten Personen in Kurzportraits vorgestellt.

Januar 2016

Die Herausgeber

Inhaltsverzeichnis

Schätzung eines Libellenbestands im Naturschutzgebiet Heiliges Meer mit der Rückfangmethode der Biostatistik	1
Claudia Cottin, Andreas Kronshage, Dimitri Kriger	
Optimierung von Produktionsprozessen in der Biotechnologie	18
Sabrina Proß	
Simulation und Optimierung der Personalplanung im Pflegebereich von Krankenhäusern durch Petri-Netz-Modelle	36
Timo Lask, Hermann-Josef Kruse, Bernhard Bachmann	
Mathematische Modellierung des Laubharkproblems als spezielle Ausprägung von allgemeinen Entsorgungsprozessen	62
Hermann-Josef Kruse	
Heuristische Lösungsstrategien für das Laubharkproblem	89
Hermann-Josef Kruse, Natalie Spent, Ralf Derdau	
Biomimetic Methods for Power Plant Optimization	117
Friedrich Biegler-König	
Applications of Shape and Topology Optimization	130
Svetozara I. Petrova	
Wer und Was ist AMMO?	153
Die Autoren	157

Schätzung eines Libellenbestands im Naturschutzgebiet Heiliges Meer mit der Rückfangmethode der Biostatistik

Claudia Cottin¹, Andreas Kronshage², Dimitri Kriger³

¹Fachbereich Ingenieurwissenschaften und Mathematik, FH Bielefeld

²LWL-Museum für Naturkunde, Außenstelle Heiliges Meer

³ELA Container GmbH

Kontakt:

claudia.cottin@fh-bielefeld.de

Zusammenfassung: Im Naturschutzgebiet „Heiliges Meer“ lebt die Libellenart *Lestes virens* (Kleine Binsenjungfer). Ökologen sind daran interessiert, die ungefähre Populationsgröße dieser Tiere zu bestimmen. Die Exemplare lassen sich nicht ohne Weiteres abzählen; es ist aber bekannt, dass sie ihren relativ begrenzten Lebensraum kaum verlassen. Somit bietet es sich an, die Populationsgröße mit der sogenannten Rückfang-Methode zu schätzen. Dabei werden Libellen eingefangen und markiert. Nachfolgend (beispielsweise am nächsten Tag) werden wiederum Libellen eingefangen und aus dem Verhältnis von erneut gefangenen markierten Libellen zu insgesamt gefangenen Libellen ein Schätzwert für die Größe der Gesamtpopulation ermittelt. Dieses mathematisch vielleicht relativ simpel erscheinende Verfahren hat jedoch viele bei weitem nicht triviale Aspekte, wenn es etwa darum geht Aussagen zur Güte des Schätzwerts zu formulieren. Im Einzelnen gibt es je nach genauen Modellannahmen auch verschiedene Varianten, solche Schätzer zu berechnen, mit spezifischen Vor- und Nachteilen. An dem Libellen-Beispiel sollen derartige Modellierungsaufgaben der statistischen Ökologie – anknüpfend an die Bachelorarbeit des drittgenannten Autors – im vorliegenden Beitrag näher beleuchtet werden.

Schlüsselwörter: Biostatistik, Rückfangmethode, Libellenbestand, *Lestes virens* (Kleine Binsenjungfer), Naturschutzgebiet Heiliges Meer

1 Grundidee der Rückfangmethode

Für den Schutz und Erhalt einer biologischen Art ist es wichtig, so viel wie möglich darüber zu erfahren. Zunächst möchte man in jedem Fall die ungefähre Populationsgröße in einem beobachteten Gebiet kennen. Ein einfaches Abzählen kommt nur für sessile, das heißt streng ortsgebundene, Arten infrage oder aber für solche, die so groß und auffällig sind, dass jedes Exemplar einzeln erfasst werden kann.

Ist dies nicht der Fall, liefert die sogenannte *Rückfang-Methode*, auch *Fang-Wiederfang-Methode* genannt, einen möglichen Ansatz zur Ermittlung der Populationsgröße. Dabei werden zunächst K Exemplare der Population eingefangen, markiert und wieder freigelassen. Nachfolgend (beispielsweise am nächsten Tag) werden nochmals stichprobenartig n Tiere eingefangen und aus dem Verhältnis von erneut gefangenen markierten Exemplaren k zu insgesamt gefangenen Exemplaren ein Schätzwert für den tatsächlichen Umfang N der Gesamtpopulation ermittelt. Wenn man davon ausgeht, dass das Verhältnis der Anzahl K der markierten Exemplare zur Gesamtanzahl N in etwa dem Verhältnis der Anzahl k der im Rückfang markierten Exemplare zum Gesamtstichprobenumfang n entsprechen sollte, also $K : N = k : n$, ergibt sich z. B. als naheliegender Schätzwert für die Größe der Gesamtpopulation der sogenannte *Petersen-Lincoln-Index*

$$\hat{N} = \frac{n \cdot K}{k}.$$

Die mathematischen Eigenschaften dieses Schätzers sowie alternative Schätzer werden in den Abschnitten 5 bis 10 diskutiert. Ferner müssen für die sinnvolle Anwendbarkeit der Schätzmethode verschiedene Voraussetzungen erfüllt sein. Insbesondere dürfen zwischen dem Anbringen der Markierungen und der Erhebung der zweiten Stichprobe keine neuen Individuen hinzukommen oder abwandern, die Markierungen dürfen nicht verloren gehen und sie dürfen die Fangwahrscheinlichkeit nicht beeinflussen. Diese Voraussetzungen sind für die *Lestes virens*-Population im Naturschutzgebiet Heiliges Meer, wie nachfolgend genauer erläutert wird, zumindest näherungsweise erfüllt.

2 Die *Lestes virens*-Population im Naturschutzgebiet Heiliges Meer

Das Naturschutzgebiet Heiliges Meer liegt in Recke im nordrhein-westfälischen Kreis Steinfurt. Es beherbergt eine große Anzahl von Pflanzen und Tieren, darunter auch die Libellenart Kleine Binsenjungfer (*Lestes virens*). Zur Ermittlung der Populationsgröße kommt ein einfaches Abzählen nicht in Frage. Die Libellen bewegen sich recht schnell, sind klein und unauffällig. Die Gefahr, einige zu übersehen oder doppelt zu zählen, ist sehr hoch. Oft verstecken sie sich auch in der Vegetation. Somit entstand die Idee, die Populationsgröße an einem Gewässer im Rahmen eines Praktikums und anschließender Bachelorarbeit mit Hilfe mathematischer Methoden zu schätzen.

Die unter 1. beschriebene Rückfangmethode erscheint mit kleineren Abstrichen als Schätzverfahren prinzipiell geeignet zu sein. Im Vergleich zu Großlibellen sind Kleinlibellen wie *Lestes virens* relativ ortstreu. Nach dem Schlupf können sie eine Reifephase im Umfeld des Gewässers verbringen. Ihre Eier legen sie im Paarungsflug am Schlupfgewässer ab. Der Schlüpfmonat ist der Juli, doch erst ab August/September sind die Tiere am Gewässer zu sehen, bis sie spätestens beim ersten Frost sterben (vgl. u. a. [Bel07]).

Beim konkreten Untersuchungsgebiet (s. Abb. 2.1 und 2.2) handelt es sich um ein größeres Gewässer auf dem Gelände des Naturschutzgebietes Heiliges Meer, die sogenannte „Blänke auf Üffings Weide“, und das direkte Umfeld, eine Wiese. Die Blänke ist etwa 1600 qm groß und wurde 1991 als Artenschutzgewässer angelegt. Das Gewässer ist geprägt von einem im Untersuchungsjahr lichten, aber zunehmenden Röhrichtbestand mit einer größeren freien Wasserfläche. An die im Gewässerumfeld liegende, stellenweise feuchte bis nasse Wiese grenzen Heide, Ackerflächen und Gehölze, die für diese Kleinlibellenart mit relativ geringen Flugdistanzen keine geeigneten Lebensräume sind. Während der Reifephase können einzelne *Lestes virens* aber durchaus auf der Heidefläche angetroffen werden ([Men10]). Das nächste für *Lestes virens* geeignete Gewässer mit einem größeren Vorkommen liegt in etwa 500 Meter Luftlinienentfernung auch im Naturschutzgebiet. Dieser Umstand gewährleistet, dass die Beobachtungen nicht wesentlich durch Ab- und Zuwanderung beeinflusst werden. Die dicht aufeinander folgenden Untersuchungszeitpunkte für die einzelnen Stichproben (s. Abschnitt 3) sorgen zusätzlich dafür, dass das Gebiet als demographisch annähernd geschlossen angesehen werden kann.



Abbildung 2.1: Das Untersuchungsgebiet am 27.8.2014 (Foto: A. Kronshage)



Abbildung 2.2: Das Naturschutzgebiet. Die Blänke ist gelb umrandet (Luftbildaufnahme aus dem Archiv der Außenstelle Heiliges Meer)

3 Die Datenerhebung

Die Datenerhebung zum *Lestes virens*-Bestand erfolgte im Rahmen einer Bachelorarbeit ([Kri15]) vorgeschalteten Praktikums im Jahr 2014. Zum Fangen und Markieren der Libellen war eine geeignete Ausrüstung nötig. Die Arbeitswerkzeuge waren: ein Insektenkescher, ein Fangglas mit Lupe, wasserfeste Markierungsstifte in den Farben schwarz und grün, Tabellenformulare und ein Bestimmungsbuch. Um ein gutes Fangergebnis zu gewährleisten, wurde die Blänke in vier verschiedene Abschnitte, A/Süd, B/Ost, C/Nord, D/West, unterteilt. Diese vier Abschnitte wurden an jedem Untersuchungstag in der gleichen Reihenfolge abgegangen und die Libellenart *Lestes virens* gefangen und markiert. Das Einhalten der Reihenfolge und das gleichmäßige Begehen der einzelnen Abschnitte sorgten dafür, dass die Fangwahrscheinlichkeit bei jeder Libelle annähernd gleich blieb.

Das Markieren fand immer nach dem gleichen Prinzip statt. Nachdem eine Libelle gefangen wurde, erfolgte die biologische Bestimmung. Im Untersuchungsgebiet flogen zur Untersuchungszeit viele andere Libellenarten. Allerdings lässt sich *Lestes virens* durch typische Merkmale wie der zwischen den Augen hinten gelblich gefärbte Kopf und das bräunliche Flügelmal, das von weißen Flügeladern begrenzt ist, gut bestimmen.

Da *Lestes virens* zwei Flügelpaare besitzt und jeder Flügel eine recht große Fläche aufweist, bot es sich an, die Libellen an den Flügeln zu markieren. Insgesamt wurden die Fänge an zwölf verschiede-

nen Tagen mit geeigneten Wetterbedingungen zwischen dem 01.09.2014 und dem 07.10.2014 (01.09., 03.09., 04.09., 05.09., 08.09., 09.09., 10.09., 16.09., 29.09., 30.09., 06.10., 07.10.) durchgeführt. Für jeden der zwölf Fangtage wurde eine andere Markierung gewählt (unterschieden durch den jeweils markierten Flügel sowie Markierungsfarbe und -symbol). Ein typischer Markierungsvorgang – nach zuvor erfolgter Artbestimmung – ist in Abb. 3.1 dargestellt. Abb. 3.2 zeigt eine wiedergefangene Libelle mit zwei verschiedenen Markierungen.



Abbildung 3.1: Markierungsvorgang (Aufnahme: D. Kriger)



Abbildung 3.2: Kleine Binsenjungfer mit grüner und schwarzer Markierung (Aufnahme: D. Kriger)

Schätzung eines Libellenbestands

Aus den Beobachtungen an den zwölf Fangtagen wurde Tabelle 3.1 erstellt, in der die Anzahl der Wiederfänge an einem bestimmten Tag bezogen auf einen bestimmten Markierungstag erfasst ist. Eine derartige Form der Datenerhebung bezeichnet man in der Fachliteratur auch als Schnabel-Census; vgl. etwa [Seb82] oder [MMK14].

Tabelle 3.1: Erfassung der Fänge und Wiederfänge

Anzahl ges.	K	43	65	54	61	62	42	46	18	32	35	9
	Wiederfang	aus Fangtag										
n	am Tag	1	2	3	4	5	6	7	8	9	10	11
65	2	1										
54	3	0	1									
61	4	4	1	4								
62	5	1	1	0	2							
42	6	2	2	0	2	1						
46	7	0	3	0	1	2	6					
18	8	0	3	0	1	1	0	0				
32	9	0	1	1	0	0	1	3	0			
35	10	1	1	0	0	0	1	4	0	2		
9	11	1	0	0	0	0	1	0	0	0	2	
5	12	0	0	0	0	0	0	0	1	1	0	0

4 Wiederfänge als Zufallsvariable

Bevor mit dem in Abschnitt 1 eingeführten Petersen-Lincoln-Index und weiteren Verfahren die Populationsgröße geschätzt wird, ist es sinnvoll, sich den Zufallscharakter der Werte aus Tabelle 3.1 klar zu machen.

Bezeichnet man mit N die (unbekannte) tatsächliche Populationsgröße, mit $K = K(t)$ die Anzahl der an einem bestimmten Fangtag t markierten Exemplare und mit $n = n(s)$ den Stichprobenumfang an einem bestimmtem Rückfangtag $s > t$, so ist die zufallsbehaftete Anzahl X der markierten Rückfänge offenbar hypergeometrisch verteilt mit

$$P(X = k) = \frac{\binom{K}{k} \cdot \binom{N-K}{n-k}}{\binom{N}{n}};$$

denn es handelt sich um das stochastische Modell des Ziehens ohne Zurücklegen aus N Exemplaren, darunter K markierte. Für Erwartungswert und Varianz von X gilt folglich

$$E(X) = n \cdot \frac{K}{N} ; \quad V(X) = n \cdot \frac{K}{N} \cdot \left(1 - \frac{K}{N}\right) \cdot \frac{N-n}{N-1}.$$

Somit handelt es sich bei dem Schätzer $\hat{p} := X/n$ um einen erwartungstreuen Schätzer für den Prozentsatz p der markierten Tiere in der Gesamtpopulation, also

$$E(\hat{p}) = \frac{K}{N} ; V(\hat{p}) = \frac{1}{n} \cdot \frac{K}{N} \cdot \left(1 - \frac{K}{N}\right) \cdot \frac{N-n}{N-1}.$$

Man beachte jedoch, dass damit $\hat{N} := n \cdot K/X$ kein erwartungstreuer Schätzer für die Gesamtpopulation ist, denn es gilt

$$E(\hat{N}) = n \cdot K \cdot E\left(\frac{1}{X}\right) \geq n \cdot K \cdot \frac{1}{E(X)} = N,$$

wobei für nicht konstantes X die echte Ungleichheitsbeziehung gilt. Die Beziehung $E(1/X) \geq 1/E(X)$ ist ein Spezialfall der Jensenschen Ungleichung und entspricht anschaulich der Tatsache, dass der arithmetische Mittelwert verschiedener Zahlen immer größer ist als der harmonische Mittelwert. Außerdem ist der Schätzwert für $X = 0$ nicht definiert bzw. würde eine „unendlich große“ Population prognostizieren.

In Tabelle 4.1 ist die hypergeometrische Verteilung von X für einige typische Werte von K bzw. n und (wie die Schätzungen in den folgenden Abschnitten nahe legen) von N in Form der auf vier Stellen gerundeten kumulierten Wahrscheinlichkeiten angegeben. Auf eine Erfassung bei unterschiedlichen Werten von K und n wurde verzichtet, da dies in der gegebenen Größenordnung kaum weitere Einsichten liefern würde.

Tabelle 4.1: Wahrscheinlichkeit für höchstens k Wiederfänge bei hypergeometrischer Verteilung

N = K = n =	1000 40	1000 50	1250 40	1250 50	1500 40	1500 50	1750 40	1750 50	2000 40	2000 50
k =	Wahrscheinlichkeit für höchstens k Wiederfänge ☞ Rechteckiges Ausschneiden									
	P(X ≤ k)									
0	0,1889	0,0720	0,2666	0,1245	0,3343	0,1784	0,3924	0,2298	0,4421	0,2775
1	0,5172	0,2717	0,6309	0,3951	0,7108	0,4967	0,7681	0,5779	0,8104	0,6425
2	0,7880	0,5375	0,8672	0,6770	0,9121	0,7692	0,9390	0,8308	0,9561	0,8728
3	0,9291	0,7636	0,9642	0,8648	0,9802	0,9184	0,9882	0,9483	0,9925	0,9658
4	0,9814	0,9017	0,9925	0,9546	0,9965	0,9771	0,9982	0,9875	0,9990	0,9928
5	0,9961	0,9662	0,9987	0,9875	0,9995	0,9948	0,9998	0,9976	0,9999	0,9988
6	0,9993	0,9903	0,9998	0,9972	0,9999	0,9990	1,0000	0,9996	1,0000	0,9998
7	0,9999	0,9976	1,0000	0,9995	1,0000	0,9998	1,0000	0,9999	1,0000	1,0000
8	1,0000	0,9995	1,0000	0,9999	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
9	1,0000	0,9999	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
10	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Die hypergeometrische Verteilung der Wiederfanganzahl X ergibt sich aus der Annahme fest vorgegebener Werte K und n . Tatsächlich sind diese Werte in der Untersuchung zwar bekannt, allerdings war die Anzahl der Fänge an den verschiedenen Erhebungstagen nicht von vornherein vorgegeben, sondern hat sich in gewissem Sinne ähnlich wie die Anzahl der Wiederfänge ebenfalls zufällig ergeben. Eine entsprechende stochastische Modellierung kann Basis für weitere Aussagen zur Schätzung von N sein, siehe etwa Abschnitt 8.

Für eine solche stochastische Modellierung betrachtet man T aufeinander folgende Stichproben vom Umfang $n_i, i = 1, \dots, T$ – im vorliegenden Anwendungsfall also $T = 12$ bzw. bei eventueller Vernachlässigung einiger Stichproben $T \leq 12$. Wie in Abschnitt 3 beschrieben werden die Elemente jeder Stichprobe mit individuellen Markierungen versehen. Nach Abschluss der Datenerhebung kennt man die Anzahl D (im Beispiel $D = 418$) der mindestens einmal markierten Exemplare. Weiterhin kennt man dazu auch die genaue Fanggeschichte aller Exemplars, das heißt, in welchen Stichproben sie beobachtet wurden, beispielsweise beschrieben durch ein T -Tupel mit Einträgen 0 oder 1, je nachdem, ob sie in der i -ten Stichprobe enthalten waren oder nicht. Schließlich sei noch a_k die Anzahl von Tieren mit der gleichen Fanggeschichte, wobei $D = \sum_k a_k$, und p_k die Wahrscheinlichkeit dieser Fanggeschichte. (Die Anzahl verschiedener Fanggeschichten ist hier der Bequemlichkeit halber nicht genauer angegeben, zumal die allgemeine Formulierung auch Varianten erlaubt, bei denen die Fangwahrscheinlichkeit nicht an jedem Beobachtungstag gleich ist.) Somit liegt nun unter der Voraussetzung, dass die einzelnen Fänge unabhängig voneinander sind, eine Multinomialverteilung vor, und für die Wahrscheinlichkeit einer bestimmten Fanggeschichte gilt also

$$P(A = a_k) = \frac{N!}{\prod_k a_k! (N - D)!} \cdot \left(1 - \sum_k p_k\right) \cdot \prod_k p_k^{a_k}.$$

Des Weiteren werden im Rahmen von Schätzungen der Populationsgröße des Öfteren auch univariate oder multivariate Poisson-Verteilungen angenommen; diese können als Approximation von hypergeometrischen Verteilungen oder Binomial- bzw. Multinomialverteilungen aufgefasst werden; vgl. dazu etwa [PBSV97].

5 Der Petersen-Lincoln-Index als Schätzwert für die Gesamtpopulation

Auf der Basis von Tabelle 3.1 kann man nun mit dem in Abschnitt 1 eingeführten Petersen-Lincoln-Index

$$\hat{N} = \frac{n \cdot K}{k}$$

Schätzwerte für die Größe der Gesamtpopulation berechnen. In dieser Schreibweise ist die Zufallsvariable X der Rückfanganzahl durch die konkrete Realisation k ersetzt, also auch \hat{N} nun ein konkreter Zahlenwert. Im Prinzip kann der Wert für jeden der Fangtage 1 bis 11 und Wiederfangtage 2 bis 12 berechnet werden, sofern nicht die Anzahl k der Wiederfänge Null ist. Das Ergebnis ist in Tabelle 5.1 dargestellt.

Man erkennt, dass die einzelnen Schätzungen stark voneinander abweichen. Die Werte kleiner als 418 (Gesamtanzahl der mindestens einmal gefangenen Libellen) sind offensichtlich unplausibel. Bei den nicht definierten Indexzahlen (mit $k = 0$) muss allein aufgrund der zugehörigen Beobachtung von einer sehr großen Population ausgegangen werden. Diese Fälle dürfen bei der Ermittlung eines durchschnittlichen Schätzwerts somit keinesfalls einfach außer Acht gelassen werden.

Tabelle 5.1: Petersen-Lincoln-Index für alle Kombinationen von Fang- und Wiederfangtagen (n/a : Wert konnte wegen $k = 0$ nicht berechnet werden)

Petersen-Lincoln-Index											
Wiederfang	aus Fangtag										
am Tag	1	2	3	4	5	6	7	8	9	10	11
2	2795										
3	n/a	3510									
4	656	3965	824								
5	2666	4030	n/a	1891							
6	903	1365	n/a	1281	2604						
7	n/a	997	n/a	2806	1426	322					
8	n/a	390	n/a	1098	1116	n/a	n/a				
9	n/a	2080	1728	n/a	n/a	1344	491	n/a			
10	1505	2275	n/a	n/a	n/a	1470	403	n/a	560		
11	387	n/a	n/a	n/a	n/a	378	n/a	n/a	n/a	158	
12	n/a	n/a	n/a	n/a	n/a	n/a	n/a	90	160	n/a	n/a

Eine einfache Durchschnittsbetrachtung könnte darin bestehen, in die Petersen-Lincoln-Formel jeweils Durchschnittswerte für n , K und k einzusetzen. Mit den Durchschnittswerten $n = 39$, $K = 42,45$ und $k = 0,84$ ergibt sich der Schätzwert $\hat{N} = 1971$. Lässt man den letzten, oder gar die letzten beiden, Untersuchungstage außer Acht, ergeben sich etwas höhere Durchschnittswerte, was auch plausibel ist, da davon ausgegangen werden kann, dass im konkreten Beispiel bei fehlenden Neugeburten schon ein Teil der ursprünglichen Population gestorben ist. Für einen etwas niedrigeren tatsächlichen Wert spricht allerdings die Tatsache, dass der Peterson-Lincoln-Index die wahre Populationsgröße tendenziell unterschätzt; vgl. Abschnitt 4. Generell erscheint das Schätzergebnis aufgrund der hohen beobachteten Schwankungen in den Stichprobenergebnissen aber als sehr unzuverlässig.

6 Der Chapman-Index als Schätzwert für die Gesamtpopulation

Da der Petersen-Lincoln-Index für $X = k = 0$ nicht definiert ist, ergibt sich die Idee, diesen zu modifizieren, indem statt mit der Zufallsvariablen X der markierten Rückfänge (siehe Abschnitt 4) mit der Zufallsvariablen $X + 1$ gearbeitet wird. Dabei liegt es nahe, Konstanten c und d zu suchen, sodass

$$\hat{N} = \frac{c}{X + 1} + d$$

Schätzung eines Libellenbestands

ein erwartungstreuer Schätzer für N ist. Man rechnet relativ leicht nach (vgl. z. B. [PBSV97]), dass dies im Fall $N - K < n$ für $c = (n + 1) \cdot (K + 1)$ und $d = -1$ der Fall ist. Der sich so ergebende Schätzwert

$$\hat{N} = \frac{(n + 1) \cdot (K + 1)}{k + 1} - 1$$

(wobei man wie beim Petersen-Lincoln-Index die Zufallsvariable X wieder durch die konkrete Realisation k ersetzt hat) ist unter dem Namen Chapman-Index bekannt.

Die Bedingung $N - K < n$ bedeutet jedoch, dass der Stichprobenumfang des Wiederfangs größer ist, als der nicht markierte Teil der Gesamtpopulation, was in der konkreten Untersuchung, wie überhaupt in den meisten realistischen Fällen, nicht zutrifft. Für $N - K \geq n$ ist der Chapman-Index jedoch nicht mehr erwartungstreu; vielmehr gilt (vgl. z. B. [PBSV97])

$$E(\hat{N}) = N - (N + 1) \cdot \frac{\binom{N-K}{n+1}}{\binom{N+1}{n+1}} \approx N \cdot \left[1 - \exp\left(-\frac{(n + 1) \cdot (k + 1)}{N}\right) \right].$$

Somit wird, genau umgekehrt wie beim Petersen-Lincoln-Index, die tatsächliche Populationsgröße durch den Chapman-Index tendenziell unterschätzt. Tabelle 6.1 zeigt den Chapman-Index für jeden der Fangtage 1 bis 11 und Wiederfangtage 2 bis 12.

Führt man eine analoge Durchschnittsbetrachtung wie beim Petersen-Lincoln-Index in Abschnitt 5 durch, erhält man den Schätzwert $\hat{N} = 945$, bzw. $\hat{N} = 1056$ oder $\hat{N} = 1151$, wenn man den letzten oder die letzten beiden Untersuchungstage außer Acht lässt. Nunmehr kann man vermuten, dass die wahre Population tatsächlich etwas größer ist.

Allerdings sind die in Tabelle 6.1 erfassten Schwankungen der Schätzwerte wieder so groß, dass Hinweise zum wahren Wert nur als recht unzuverlässig erachtet werden können. In [PBSV97] wird ausgeführt, dass für eine sinnvolle Anwendung des Petersen-Lincoln- bzw. des Chapman-Index möglichst $n \cdot K > 3N$ sein sollte, was für die geschätzten Werte von N nicht annähernd zutreffend wäre.

Tabelle 6.1: Chapman-Index für alle Kombinationen von Fang- und Wiederfangtagen

Chapman-Index											
Wiederfang	aus Fangtag										
am Tag	1	2	3	4	5	6	7	8	9	10	11
2	1451										
3	2419	1814									
4	545	2045	681								
5	1385	2078	3464	1301							
6	630	945	2364	888	1354						
7	2067	775	2584	1456	986	288					
8	835	313	1044	588	598	816	892				
9	1451	1088	907	2045	2078	709	387	626			
10	791	1187	1979	2231	2267	773	337	683	395		
11	219	659	549	619	629	214	469	189	329	119	
12	263	395	329	371	377	257	281	56	98	215	59

7 Petersen-Lincoln- und Chapman-Index für Dreifach-Fänge

Eine genauere Analyse der vorliegenden Daten zeigt, dass insgesamt sechs Tiere sogar dreimal gefangen wurden. Somit kann der Petersen-Lincoln-Index und der Chapman-Index auch für diese Dreifach-Fänge berechnet werden. Dazu sei nun K die Gesamtzahl der bis zum Tag 11 zweifach markierten Tiere und n die durchschnittliche Anzahl der nach einer Doppelmarkierung vollzogenen weiteren Fänge. Dann erhält man mittels

$$\hat{N} = \frac{n \cdot K}{6} \text{ bzw. } \hat{N} = \frac{(n+1) \cdot (K+1)}{7} - 1$$

einen modifizierten Petersen-Lincoln- bzw. Chapman-Index bezogen auf die Dreifach-Fänge. In der vorliegenden Untersuchung ist $K = 46$ und $n = 150,3$ (die genauen Daten, die zu diesen Werten führen, können auf Anfrage bereitgestellt werden) und somit $\hat{N} = 1152$ bzw. $\hat{N} = 1015$. Die Schätzwerte gemäß Petersen-Lincoln- bzw. Chapman-Index liegen nun weit weniger auseinander als in den Berechnungen von Abschnitt 5 und 6. Auch kann nun die Bedingung $n \cdot K > 3N$ für die Angemessenheit des Schätzverfahrens als erfüllt angesehen werden.

Ähnlich wie zuvor für den Petersen-Lincoln- bzw. Chapman-Index erläutert, könnte man vermuten, dass auch hier der erste Schätzwert eher zu hoch und der zweite eher zu niedrig gegriffen ist. Allerdings ist anzumerken, dass die obige Konstruktion des Schätzers unter Einbeziehung der Dreifach-Fänge zunächst eher heuristischer Natur ist und Aussagen zur Güte noch umfangreichere mathematische Zusatzüberlegungen erfordern würden. Außerdem sollte beachtet werden, dass die Beobachtung von Dreifach-Fängen unter anderem auch wegen der Sterblichkeit der Tiere im Zeitverlauf recht selten sein könnte. Diese Tatsache würde wiederum eher für einen höheren tatsächlichen Wert von N sprechen; siehe dazu auch Abschnitt 9.

8 Der sequenzielle Schätzansatz

Die in den Abschnitten 5 und 6 beobachtete starke Streuung der Schätzwerte tritt vor allem deshalb auf, weil die Anzahl markiert wiedergefangener Exemplare sehr klein ist. In dieser Situation bietet sich alternativ ein sequenzieller Schätzansatz an. Dabei wird anschließend an einen bestimmten Markierungstag solange weitergefangen, bis man erstmalig auf ein Tier mit dieser Markierung trifft bzw. allgemeiner bis man s markierte Individuen wiedergefangen hat.

Die zufallsbehaftete Anzahl Y_m von Wiederfängen, bis erstmalig m markierte Individuen gefunden wurden, besitzt die Wahrscheinlichkeitsverteilung

$$P(Y_m = k) = \frac{\binom{K}{m-1} \cdot \binom{N-K}{m-k}}{\binom{N}{k-1}} \cdot \frac{K-m+1}{N-k+1}, \quad k = m, m+1, \dots, m+N-K.$$

Dabei steht der linke Bruch für die hypergeometrische Wahrscheinlichkeitsverteilung, dass unter $k-1$ Tieren aus der Gesamtanzahl N genau $m-1$ markierte sind (dies entspricht der sequenziellen Vorgehensweise vor Fang des m -ten markierten Exemplars), und der rechte Bruch für die Wahrscheinlichkeit, aus einem restlichen Bestand von $N-k+1$ unter dann noch $K-m+1$ markierten Individuen ein weiteres markiertes Exemplar zu finden. Aus Y_m ergibt sich gemäß

$$\hat{N}_m = \frac{K+1}{m} \cdot Y_m - 1$$

ein erwartungstreuer Schätzer für N mit Varianz

$$V(\hat{N}_m) = \frac{(K-m+1) \cdot (N+1) \cdot (N-K)}{m \cdot (K+2)},$$

die Varianz nimmt also, anschaulich auch plausibel, mit wachsendem m ab. Für weitere Einzelheiten zu diesem Schätzer \hat{N}_m und zur obigen Herleitung siehe etwa [PBSV97].

Das vorhandene Datenmaterial ermöglicht die Angabe von Y_m für $m = 1, \dots, 4$ und die ersten sieben Markierungstage. Diese Werte und die darauf basierenden Schätzergebnisse sind in Tabelle 8.1 dargestellt. Die Durchschnittswerte für die Schätzer sind mit der Schätzformel aus diesem Abschnitt auf Basis der Durchschnittswerte von K und Y_m berechnet.

Tabelle 8.1: Schätzwerte für die Populationsgröße im sequenziellen Ansatz

Tag	K	Y_1	Y_2	Y_3	Y_4	N_1^	N_2^	N_3^	N_4^
1	43	44	134	142	157	1935	2947	2082	1726
2	65	24	109	133	198	1583	3596	2925	3266
3	54	1	7	16	53	54	192	292	728
4	61	19	30	72	87	1177	929	1487	1348
5	62	2	64	74	94	125	2015	1553	1480
6	42	2	4	8	18	85	85	114	193
7	46	25	42	46	61	1174	986	720	716
Durchschnitt	53,3	16,7	55,7	70,1	95,4	906	1511	1268	1294

Geht man nun beispielsweise von dem Durchschnittswert $K = 53$ und einer tatsächlichen Populationsgröße $N = 1200$ aus, was größenordnungsmäßig in den Rahmen der bisher berechneten Schätzungen passt, so erhalte man für die Varianz der Schätzer die Ergebnisse

$$V(\hat{N}_1) = 1.327.454; \quad V(\hat{N}_2) = 651.204; \quad V(\hat{N}_3) = 425.787; \quad V(\hat{N}_4) = 313.078$$

mit zugehörigen Standardabweichungen (Wurzel aus der Varianz)

$$\sigma(\hat{N}_1) = 1152; \quad \sigma(\hat{N}_2) = 807; \quad \sigma(\hat{N}_3) = 653; \quad \sigma(\hat{N}_4) = 560.$$

Eine derartige Streuung der Schätzwerte spiegelt sich größenordnungsmäßig auch in Tabelle 8.1 wider. Auch bei den sequenziellen Schätzverfahren ist die mögliche Streuung der Schätzwerte also groß und die berechneten Schätzwerte entsprechend unzuverlässig. Allerdings bewegt man sich wie zuvor auch hier in einer geschätzten Größenordnung von um die bzw. wahrscheinlich etwas über 1000 Exemplare, für die die verschiedenen Auswertungen sprechen.

9 Maximum-Likelihood-Schätzer

Die Idee von Maximum-Likelihood-Schätzern (ML-Schätzern) besteht allgemein gesprochen darin, zu einem vorgegebenen Wahrscheinlichkeitsmodell die Parameter so zu schätzen, dass gerade ihr Auftreten am wahrscheinlichsten ist.

Nimmt man beispielsweise an, dass die zufallsbehaftete Anzahl X der markierten Rückfänge nicht hypergeometrisch, sondern vielmehr binomial verteilt mit Stichprobenumfang n und Wiederfangwahrscheinlichkeit $p = K/N$ ist (dies entspricht der stochastischen Modellierung von K und n bei lediglich zwei Fangtagen; vgl. die Ausführungen in Abschnitt 4), so ergibt sich als ML-Schätzer für N der Petersen-Lincoln-Index. Das bedeutet also, dass mit $p = K/N$ die Wahrscheinlichkeit

$$P(X = k) = \binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k}$$

Schätzung eines Libellenbestands

gerade für $N = \hat{N} = \frac{n \cdot K}{k}$ maximal wird. Dies erkennt man durch Ableiten und Nullsetzen von $P(X = k)$ aufgefasst als Funktion von N ; ausführlicher vgl. etwa [Man15].

Allgemeiner kann man die Maximum-Likelihood-Methode auch anwenden, um auf der Basis der Information zu allen Fangtagen und der in Abschnitt 4 beschriebenen Multinomialverteilung einen Schätzer für N zu konstruieren. Bezeichnet man in diesem Zusammenhang nun die am i -ten Beobachtungstag gefangenen Exemplare mit n_i , so ergibt sich für N die Bestimmungsgleichung (vgl. etwa [Seb82] oder [Man15])

$$1 - \frac{D}{N} = \prod_{i=1}^T \left(1 - \frac{n_i}{N}\right).$$

Wie zuvor steht hier T für die Anzahl der Beobachtungstage und D für die Anzahl verschiedener gefangener Tiere insgesamt.

Die Bestimmungsgleichung ist äquivalent zur Aufgabe einer Nullstellenbestimmung für ein Polynom T -ten Grades in N , die bekanntlich im Allgemeinen nur numerisch gelöst werden kann. Unter Verwendung von MAPLE wurde mit den Werten aus Tabelle 3.1 der Wert $N = 1721$ berechnet. Lässt man die letzten beiden Beobachtungstage mit nur wenigen Fängen weg, erhält man $N = 1823$. Die Schätzwerte sind also etwas niedriger als beim einfachen Petersen-Lincoln-Index.

Wie der Petersen-Lincoln-Index (Fall $T = 2$) ist auch der allgemeinere ML-Schätzer nicht erwartungstreu. Man kann allgemeinere mathematische Erkenntnisse über ML-Schätzer ausnutzen, um präzisere Aussagen zum Schätzer zu erhalten, wie etwa zu dessen Verzerrung (Abweichung vom Erwartungswert) und zur Varianz. Grundsätzlich gesprochen ist letztere im vorliegenden Beispiel leider – hinsichtlich der Zuverlässigkeit der Schätzwerte – wieder sehr groß, sodass an dieser Stelle auf genauere Angaben verzichtet werden und dazu nur auf die weiterführende Literatur (etwa [Seb82], [Man15], [MMK14]) verwiesen werden soll.

10 Ausblick auf weitere, insbesondere simulationsbasierte, Schätzverfahren

In weiterführender Literatur wie [Seb82] oder [MMK14] finden sich viele Hinweise auf die Konstruktion von Schätzern unter allgemeineren Voraussetzungen als in dieser Studie, etwa unter Berücksichtigung von Geburts- und Todesprozessen, Wanderungsbewegungen oder unterschiedlichen Fangwahrscheinlichkeiten für verschiedene Beobachtungstage. Eine weitere Modifikation besteht in der Verwendung anderer Verteilungsmodelle, beispielsweise einer Poisson-Verteilung für die Anzahl der Wiederfänge (s. auch Abschnitt 4). Weiterhin besteht ggf. die Möglichkeit gewisse Vorabinfor-

mationen, etwa Erkenntnisse aus vergangenen Jahren, mit in die Schätzungen einzubeziehen; man spricht dabei auch von Bayes-Schätzern.

Als konkretes Schätzverfahren wird auch unter solchen allgemeineren bzw. modifizierten Ansätzen oft die ML-Methode verwendet. Nur in Spezialfällen wird es allerdings möglich sein, explizit die komplette Wahrscheinlichkeitsverteilung für einen Schätzer \hat{N} , oder auch nur charakteristische Parameter wie etwa die Varianz der Verteilung, anzugeben und selbst wenn, hängen diese in der Regel von der unbekanntem wahren Anzahl N ab; siehe etwa Abschnitt 4.

Um einen besseren Eindruck von der Wahrscheinlichkeitsverteilung von \hat{N} zu bekommen und auf dieser Basis ggf. neue Schätzwerte zu berechnen, kann man sich stochastischer Simulationen bedienen. Da dies allerdings einigen Implementierungs- und Rechenaufwand erfordert, wurden derartige Berechnungen im Rahmen der vorliegenden Untersuchung noch nicht durchgeführt. Es soll nachfolgend nur grob erläutert werden, wie solche Verfahren grundsätzlich funktionieren. Zur grundsätzlichen stochastischen Modellbildung dafür vergleiche Abschnitt 4.

Zunächst erzeugt man eine große Anzahl hypergeometrisch verteilter (oder gemäß einem anderen geeigneten Verteilungsmodell verteilter) Zufallszahlen mit Verteilungsparametern N , K und n . Diese Zufallszahlen stehen für die zufällige Anzahl von Wiederfängen an einem bestimmten Tag, stellen also viele Simulationen des realen Wiederfangprozesses dar. Auf dieser Grundlage berechnet man wiederum viele Schätzwerte \hat{N} mit einer der zuvor angesprochenen Schätzmethoden. Als Ausgangswert für N kann man einen der zuvor (nicht simulationsbasiert) berechneten Schätzwerte ansetzen. Für K und n wählt man die tatsächlich beobachteten Werte, oder aber man modelliert diese ebenfalls als Zufallsvariablen, wie in Abschnitt 4 erläutert, ggf. auch unter den weiter oben angesprochenen allgemeineren Modellannahmen.

Somit erhält man eine empirische Wahrscheinlichkeitsverteilung für \hat{N} . Diese liefert einerseits genauere Einblicke in die Schwankungsbreite der möglichen Schätzwerte und kann andererseits selbst Grundlage für weitere Schätzungen sein, indem beispielsweise ein Mittelwert der Verteilung als neuer Schätzwert für N gewählt wird. Bei solchen simulationsbasierten Schätzern spricht man auch von Bootstrap-Schätzern. Exemplarische Beispiele von Fang-Wiederfang-Simulationen findet man beispielsweise in [RS90] oder [EG08].

11 Zusammenfassung und Fazit

In der vorliegenden Untersuchung wurden am Beispiel der Schätzung eines Libellenbestands verschiedene auf der sogenannten Rückfangmethode basierende Schätzverfahren erläutert. Aus mathematischer Sicht bestand das Ziel hauptsächlich darin, grundlegende Aspekte auf exemplarische

Schätzung eines Libellenbestands

Weise derart darzustellen, dass sie auch auf ähnlich gelagerte statistische Fragestellungen angewendet und dann vor allem auch richtig interpretiert werden können. Naheliegender ist die Schätzung anderer Tierbestände. Die Rückfang-Methode wird aber beispielsweise auch in der Meinungsforschung oder bei soziodemografischen Erhebungen eingesetzt. (Die „Markierung“ entspricht dann der Registrierung eines bestimmten Personenkreises, deren Mitglieder unter Umständen erneut befragt bzw. beobachtet werden.)

Das konkrete Ergebnis der Untersuchungen war aber auch aus ökologischer Sicht interessant. Bei allen beschriebenen Unwägbarkeiten kann man schließen, dass die Kleine Binsenjungfer im Jahr 2014 mit einem relativ großen Bestand an der Blänke vorkam. Zurzeit ist das Gewässer für diese Art ein idealer Lebensraum. Vor einigen Jahren wurde unter anderem dieses Gewässer im Naturschutzgebiet Heiliges Meer auch von [Men10] untersucht. Für die Jahre 2008 bis 2010 sind neun Libellenarten nachgewiesen. Es fehlt aber in diesem Zeitraum *Lestes virens*. Im Jahr 2010 befand sich das Gewässer in einem schlechten Zustand, da sich das Röhricht im gesamten Gewässer sehr dicht ausgebreitet hatte. In 2011 wurde die Blänke deshalb durch Abschieben des Röhrichts renaturiert und an einer Stelle vertieft mit offenem Wasserbereich. Nur ein kleiner Röhrichtbestand blieb übrig. Durch diese Maßnahme wurde das Gewässer in den Folgejahren nicht nur für *Lestes virens* wieder attraktiv. Auch in 2015 wurden viele Kleine Binsenjungfern nachgewiesen (Ch. Göcking / Libellenkurs Heiliges Meer, schriftl. Mitteilung, 8.10.2015). Das Röhricht breitet sich seit der Abschiebung wieder aus, ist aber noch relativ licht.

Literaturverzeichnis

- [Bel07] BELLMANN, H.: Der Kosmos Libellenführer. Franckh-Kosmos-Verlag, 2007.
- [EG08] ENGEL, J. und GRÜBEL, R.: Bootstrap – oder die Kunst, sich selbst aus dem Sumpf zu ziehen. Math. Semesterber., (55):113–130, 2008.
- [Kri15] KRIGER, D.: Die Fang-Wiederfang-Methode der Biostatistik erläutert an populationsdynamischen Untersuchungen zur Libellenart *Lestes virens*. Bachelorarbeit im Studiengang Angewandte Mathematik der FH Bielefeld, 2015.
- [Man15] MANNER, H.: Stichprobenverfahren. Skript, Fakultät Statistik, Technische Universität Dortmund, Sommersemester 2015. (Web-Link: https://www.statistik.tu-dortmund.de/fileadmin/user_upload/Skript_SoSe_2015.pdf; Abrufdatum 13.09.2015).
- [Men10] MENKE, N.: Bestandserfassung der FFH-Libellenarten im NSG Heiliges Meer 2010. Unveröffentlichtes Gutachten im Auftrag des LWL-Museums für Naturkunde, Münster, 30 S., 2010.

- [MMK14] MCCREA, R. S., MORGAN, B. J. T. und KÖTTER, U.: Analysis of Capture-Recapture Data. Chapman & Hall, 2014.
- [PBSV97] PFEIFER, D., BÄUMER, H.-P., SCHLEIER, U. und VALK, V. DE: Grundzüge der Statistischen Ökologie. Skript, Fakultät V, Mathematik- und Naturwissenschaften, Carl von Ossietzky Universität Oldenburg,, September 1997. (Web-Link: <http://www.staff.uni-oldenburg.de/dietmar.pfeifer/Skripten/Stat0ek.pdf>; Abrufdatum 13.09.2015).
- [RS90] RICHTER, O. und SÖNDGERATH, D.: Parameter Estimation in Ecology: The Link Between Data and Models. VCH, 1990.
- [Seb82] SEBER, G. A. F.: The Estimation of Animal Abundance. Charles Griffin & Company Ltd, 2 Auflage, 1982.

Optimierung von Produktionsprozessen in der Biotechnologie

Sabrina Proß¹

¹Fachbereich Ingenieurwissenschaften und Mathematik, FH Bielefeld

Kontakt:

sabrina.pross@fh-bielefeld.de

Zusammenfassung. Die industrielle Biotechnologie strebt bei ihren Produktionsprozessen nach einer hohen Produktausbeute bei gleichzeitig geringen Prozesskosten. Um diese konträren Ziele zu erreichen, wird in diesem Artikel eine allgemeine Vorgehensweise vorgestellt. Es wird aufgezeigt, wie das zugehörige mathematische Optimierungsproblem formuliert wird und welche Algorithmen sich zur Lösung eignen. Zudem wird ein generelles Konzept zur Etablierung einer (automatisierten) Steuerung bzw. Regelung des Bioprozesses angegeben. Alle präsentierten Konzepte werden anhand von zwei Beispielen, die Xanthanproduktion des Bakteriums *Xanthomonas campestris* und die Antikörperproduktion der CHO-Zellen, verdeutlicht.

Schlüsselwörter. Evolutionsstrategie, hybride Optimierungsverfahren, Steuerung, Regelung, Automatisierung, *Xanthomonas campestris*, CHO-Zellen

1 Einleitung

Viele Produkte werden in der Biotechnologie in **Bioreaktoren**, auch Fermenter genannt, produziert. Ein Bioreaktor ist ein Behältnis, in dem Mikroorganismen herangezüchtet werden (siehe Abbildung 1.1). Bei Mikroorganismen handelt es sich um Kleinstlebewesen, z.B. Bakterien. Das Bakterium *Xanthomonas campestris* beispielsweise produziert den Stoff **Xanthan**, der in der Industrie zahlreiche Einsatzgebiete findet. Xanthan wird als Verdickungsmittel, Emulgator¹ und Stabilisator eingesetzt. In der Lebensmittelindustrie ist Xanthan ein Bestandteil von zahlreichen Produkten, wie Salatdressings, Sirups, Suppen, Backware und Tiefkühlkost. Aber auch in der Kosmetik- und Pharmaindustrie findet Xanthan in Cremes, Lotionen, Shampoos und Zahnpasta Verwendung [GSCG00].

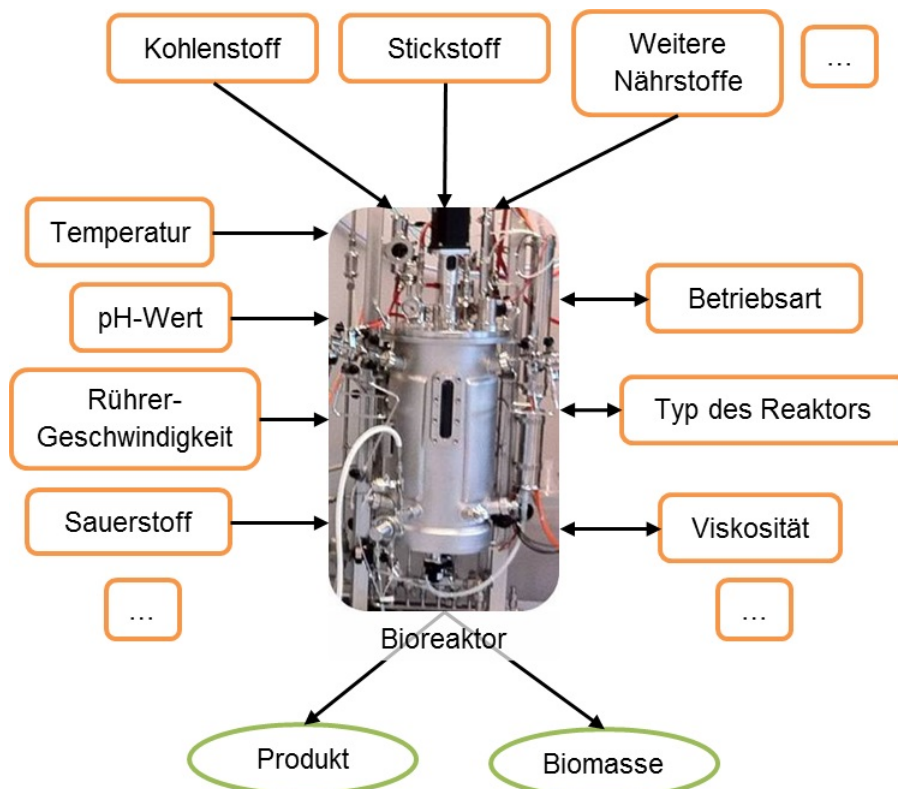


Abbildung 1.1: Einflussfaktoren auf Wachstum und Produktion der Mikroorganismen im Bioreaktor

Die Xanthanproduktion und das Wachstum der Bakterien im Bioreaktor werden von zahlreichen Faktoren beeinflusst. Abbildung 1.1 stellt einige dieser Faktoren dar. Je wohler sich die Bakterien fühlen, umso besser wachsen sie und umso mehr Xanthan wird produziert. Hieraus ist leicht ersichtlich, dass die Industrie ein großes Interesse hat diese Faktoren optimal einzustellen, um eine maximale Xanthanausbeute zu erlangen.

¹Ein Emulgator ist ein Stoff, mit dem es möglich ist zwei nicht mischbare Flüssigkeiten (z.B. Öl und Wasser) zu einem Gemisch (der sog. Emulsion) zu vermengen.

Da die Anzahl der Einstellungsparameter sehr hoch ist, müssten zur Auffindung des optimalen Parametersatzes unzählige Experimente durchgeführt werden. Diese Experimente sind aufwendig, langwierig und teuer. Ein Modell des Produktionsprozesses und eine anschließende Optimierung der Prozessparameter können teure Experimente vermeiden.

Ein weiteres Beispiel sind die CHO-Zellen (vom engl. Chinese Hamster Ovary Cells), deren produzierte **Antikörper** sich in vielen Medikamenten wieder finden [BR06]. Auch hier ist die Pharmaindustrie daran interessiert die Antikörperproduktion zu maximieren. Erschwerend kommt es hierbei jedoch zu einer Produktion von sogenannten Abfallprodukten (Laktat und Ammonium, siehe Abbildung 1.2), die das Wachstum und die Antikörperproduktion der Zellen hemmen können [LT97]. Die Zellen vergiften sich hierbei selbst und sterben. Es wird jedoch vermutet, dass die CHO-Zellen diese Abfallprodukte auch wieder aufnehmen können und zur Produktion von Antikörpern verwenden können, wenn die Umweltbedingungen entsprechend sind [PBAS06]. Dies muss in einem Modell zur Auffindung des optimalen Parametersatzes unbedingt berücksichtigt werden.

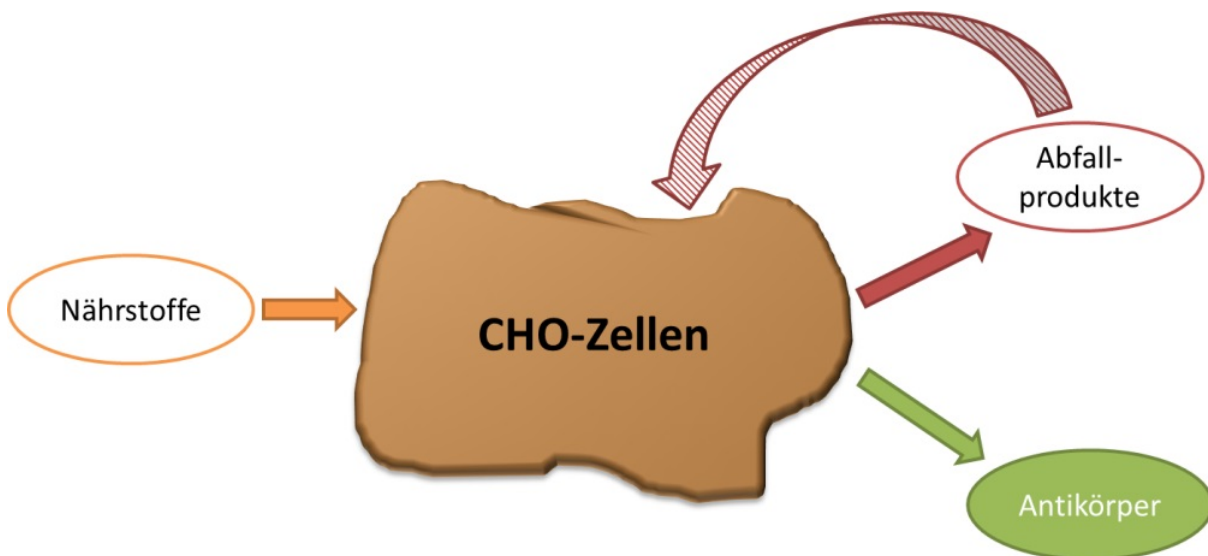


Abbildung 1.2: CHO-Zellen als Antikörperproduzent für pharmazeutische Produkte

In diesem Artikel wird eine Methodik aufgezeigt, wie sich aus einem mathematischen Modell eines biologischen Produktionsprozesses der optimale Parametersatz für die Betreibung des Bioreaktors bestimmen lässt. Auf die Modellbildung selbst wird nur kurz eingegangen und für eine detailliertere Beschreibung des Modellierungsprozesses wird auf [Pro13] verwiesen. Die Zusammenhänge werden in Abbildung 1.3 darstellt.

Zunächst wird in Abschnitt 2 das mathematische Optimierungsproblem aufgestellt, zu dem in Abschnitt 3 Algorithmen zur Lösung vorgestellt werden. In Abschnitt 4 wird anschließend aufgezeigt, wie eine Steuerung bzw. Regelung des Bioprozesses auf Grundlage der Ergebnisse der Optimierung etabliert werden kann.

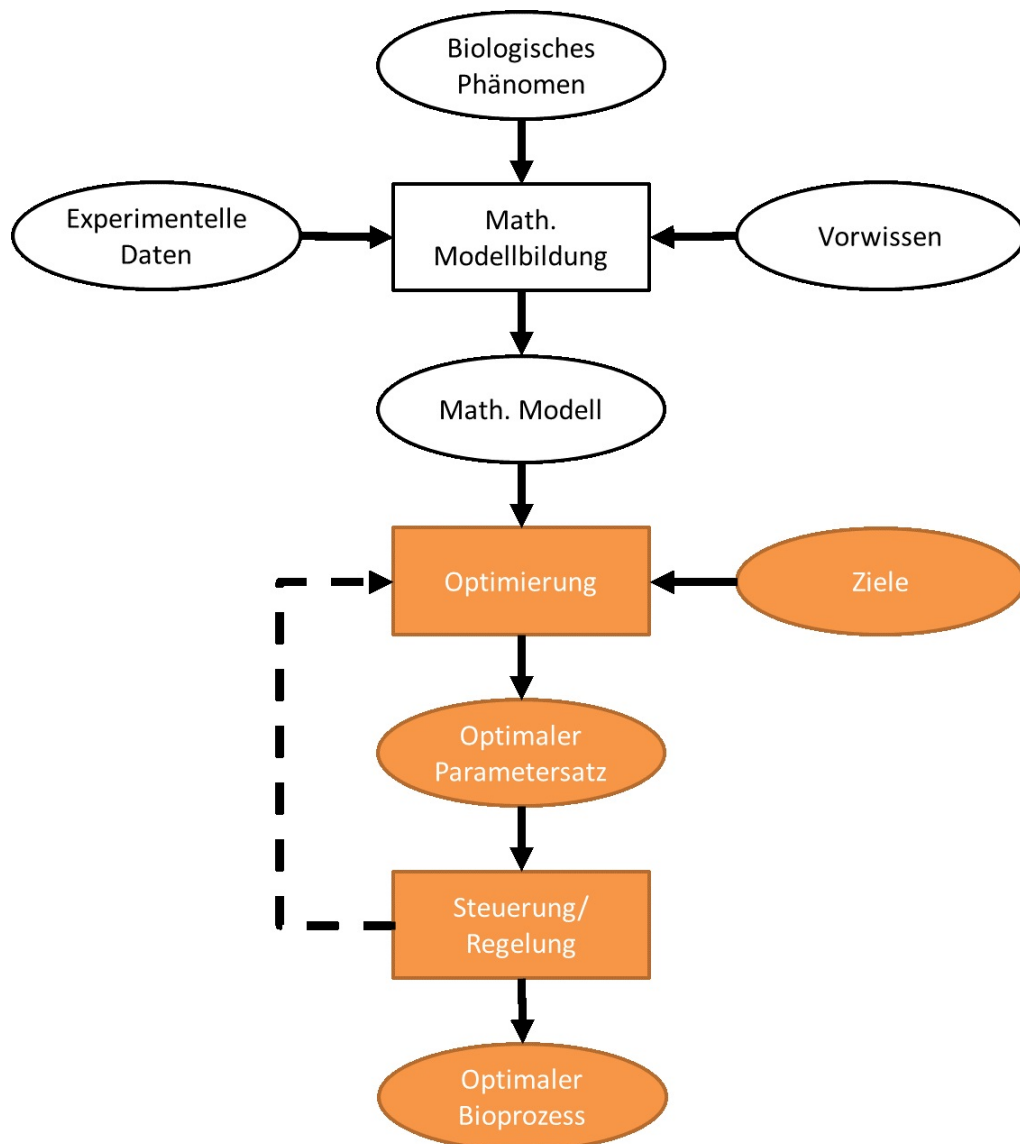


Abbildung 1.3: Der Modellierungsprozess: vom biologischen Phänomen zur optimalen Steuerung/Regelung des Produktionsprozess

2 Das mathematische Optimierungsproblem

Die konkrete Problemstellung lautet:

Maximiere die Produktausbeute zu einem bestimmten Zeitpunkt t_f während eines Fementationsprozesses. Hierbei soll die Produktausbeute von verschiedenen Prozessparametern beeinflusst werden (siehe Abbildung 1.1).

Um dieses Optimierungsproblem zu lösen, muss zunächst ein **mathematisches Modell** aufgestellt werden, dass den Produktbildungsprozess des Mikroorganismus zeitabhängig beschreibt. Zahlreiche Modellformalisen stehen zur Modellierung biologischer Systeme zur Verfügung (siehe z.B. [Wie02]). Sie unterscheiden sich in ihrer Genauigkeit und ihrer Komplexität, mit der sie den

betrachteten Organismus abbilden. Differentialgleichungssysteme und Petri-Netze sind zwei Möglichkeiten ein mathematisches Modell für einen biologischen Prozess zu erhalten (vgl. [Pro13]). Der Einfluss aller Parameter, die anschließend optimiert werden sollen, muss sich auch im Modell widerspiegeln. Soll beispielsweise die Temperatur des Bioreaktors optimiert werden, so muss der Einfluss der Temperatur auf die einzelnen Stoffwechselprozesse, auf das Wachstum und auf das Produkt (z.B. Xanthan oder Antikörper) im Modell integriert sein.

Anschließend muss auch der Bioreaktor mathematisch abgebildet werden. Ein wichtiger Parameter für die optimale Produktausbeute ist hierbei die **Betriebsart** des Bioreaktors. Grundsätzlich wird zwischen einer diskontinuierlichen (Batch, Repeated-Batch oder Fed-Batch) und einer kontinuierlichen Betriebsart unterschieden (vgl. z.B. [Chm11]). Beim **Batch-Betrieb** werden nur zu Beginn Nährstoffe¹ den Mikroorganismen zur Verfügung gestellt, wohingegen beim **Repeated-Batch** zu bestimmten Zeitpunkten weitere Nährstoffe hinzugegeben werden. Wenn beim **Fed-Batch** die Nährstoffe erstmals wachstumslimitierende Werte erreichen, werden frische Nährstoffe kontinuierlich zugeführt. Während eines **kontinuierlichen Fermentationsprozesses** wird fortlaufend flüssiges Medium mit Nährstoffen zugegeben und dieselbe Menge gleichzeitig aus dem System abgezogen.

Das mathematische Optimierungsproblem (**MOP**) kann dann wie folgt beschrieben werden

$$\max Q(\mathbf{z}, t_f) \quad \mathbf{z} \in Z \subset \mathbb{R}^n, \quad Q: Z \rightarrow \mathbb{R}$$

u.d.N.

Mathematisches Modell des Produktionsprozesses

$$\mathbf{z}^u \leq \mathbf{z} \leq \mathbf{z}^o$$

mit

- $Q(P(\mathbf{z}, t_f), t_f)$, bezeichnet als $Q(\mathbf{z}, t_f)$, als Zielfunktion, die maximiert werden soll,
- $\mathbf{z} = (z_1, z_2, \dots, z_n)$ als Prozessparameter, die variiert werden sollen, um die Produktausbeute zu maximieren,
- t_f als Zeitpunkt zu dem die Produktausbeute betrachtet wird (Dieser Zeitpunkt kann als zusätzlicher Parameter mit in die Zielfunktion aufgenommen werden, da die Produktionszeit mit den Kosten des Prozesses korreliert.),

¹Nährstoffe, auch Substrate genannt, werden von Mikroorganismen zum Wachstum und zur Produktion benötigt. Zucker sind die häufigsten Nährstoffe, aber auch Stickstoff- und Phosphatquellen werden benötigt.

²Untere und obere Grenzen für die Prozessparameter müssen nicht zwingend festgelegt werden. Da die einzelnen Parameter aber meist extrem verschiedene Größenordnungen haben, ist die Festlegung für ein Optimierungsverfahren sehr hilfreich.

- $\mathbf{z}^u = (z_1^u, z_2^u, \dots, z_n^u)$ als untere Grenze der Prozessparameter und
- $\mathbf{z}^o = (z_1^o, z_2^o, \dots, z_n^o)$ als obere Grenze der Prozessparameter.

Mögliche Zielfunktionen sind beispielsweise:

$$Q_1 = P(\mathbf{z}, t_f^{fix}) \quad \mathbf{z} \in Z \subseteq \mathbb{R}^n$$

$$Q_2 = \frac{P(\mathbf{z}, t_f)}{t_f} \quad \mathbf{z} \in Z \subseteq \mathbb{R}^n.$$

Die erste Zielfunktion betrachtet nur die Produktausbeute zu einem festen Zeitpunkt t_f^{fix} , während die zweite auch noch die dafür benötigte Zeit miteinbezieht. Da die Produktionszeit meist mit den Kosten des Produktionsprozesses korreliert, versucht die zweite Zielfunktion einen Kompromiss zwischen Produktausbeute und Kosten zu finden.

3 Lösungsverfahren für das Optimierungsproblem

Die Zielfunktion in Kombination mit dem Modell für das biologische System hat meist die unerwünschten Eigenschaften: nichtlinear, nicht stetig und nicht differenzierbar. Die Nichtlinearität ergibt sich z.B. aus der Tatsache, dass Stoffwechselreaktionen in einem Organismus desto schneller ablaufen je mehr von dem Ausgangsstoff zur Verfügung steht. Dieses Verhalten kann mit nichtlinearen Gleichungen beschrieben werden, z.B. mit der Michaelis-Menten-Kinetik (siehe z.B. [Chm11]). Unstetigkeiten treten beispielsweise immer dann auf, wenn der Bioreaktor im Repeated-Batch-Modus betrieben wird, da hierbei zu bestimmten Zeiten eine bestimmte Menge an Nährstoffen dem System (diskret) hinzugefügt wird. Zu nicht differenzierbaren Stellen kommt es u.a. wenn ein bestimmter Stoff schlagartig nicht mehr vom Organismus produziert, sondern konsumiert wird oder umgekehrt. Dies ist beispielsweise bei den eingangs vorgestellten CHO-Zellen der Fall. Werden im Bioreaktor bestimmte Bedingungen erreicht, schalten die Zellen die Produktion von den Abfallstoffen Laktat und Ammonium ab und konsumieren diese Stoffe nun und nutzen sie zur Produktion der Antikörper.

Die Nichtdifferenzierbarkeit verhindert den Einsatz von Methoden, die die Aufstiegsrichtung aus den Ableitungen der Zielfunktion bestimmen, deshalb wird bei der Auffindung des optimalen Parametersatzes auf ableitungsfreie Methoden zurückgegriffen. Dabei wird zwischen lokalen, globalen und hybriden Methoden unterschieden. **Lokale Methoden** versuchen das Maximum ausgehend von einem vorgegebenen Startpunkt zu finden. Hierbei wird nur die lokale Information über die Zielfunktion in der Nachbarschaft des aktuellen Punktes benutzt. Diese Methoden konvergieren meist zu dem lokalen Maximum, das am nächsten am Startpunkt liegt. **Globale Methoden** hingegen erforschen den gesamten Suchraum, um das globale Maximum zu finden. Lokale und globale Methoden können auch zu sogenannten **hybriden Methoden** kombiniert werden. Diese sollen

den hohen Rechenaufwand von globalen Methoden vermeiden und gleichzeitig die vorzeitige Konvergenz in einem lokalen Maximum verhindern.

Nachfolgend wird zunächst eine lokale Optimierungsmethode, die Nelder-Mead-Simplex-Methode, vorgestellt und anschließend, die globale Evolutionsstrategie ausführlich dargestellt. Anschließend wird beschrieben, wie die Vorteile beider Methoden sich zu einer hybriden Optimierungsmethode kombinieren lassen.

3.1 Nelder-Mead-Simplex-Methode

Eine lokale Optimierungsmethode, die für derartige Probleme eingesetzt werden kann, ist beispielsweise die Nelder-Mead-Simplexmethode ([NM65], siehe auch [LRWW99]). Jede Iteration startet mit einem Simplex, das durch $n + 1$ Eckpunkte festgelegt wird. Jeder Eckpunkt entspricht einem Parametersatz dessen Güte (Produktausbeute) über die Zielfunktion berechnet werden kann. Unter diesen Punkten werden der Beste und der Schlechteste ermittelt, also der Parametersatz mit der höchsten und mit der niedrigsten Produktausbeute. Der Schlechteste wird durch einen neuen Punkt ersetzt, wodurch ein neues Simplex generiert wird. Der Beste wird als derzeit beste Lösung immer beibehalten. Die Nelder-Mead-Simplexmethode ist eine der populärsten Optimierungsmethoden und ist Bestandteil von vielen numerischen Softwarepaketen wie Matlab. Dennoch gibt es bis jetzt keinen generellen Beweis für die Konvergenz des Verfahrens.

3.2 Evolutionsstrategie

Eine globale Methode ist die von Rechenberg und Schwefel in den frühen 1970ern Jahren entwickelte Evolutionsstrategie [Rec71], [Sch75]. Diese Methode gehört zur Klasse der Evolutionsalgorithmen, deren generelle Vorgehensweise in Abbildung 3.1 dargestellt wird. Hierbei handelt es sich um naturinspirierte Optimierungsverfahren, die die Prinzipien der biologischen Evolution nachahmen. Die biologische Evolution basiert auf der Theorie von Charles Darwin und erklärt die Vielfalt und Komplexität der Lebewesen. Durch zufällige Variation entstehen Individuen mit vorteilhaften Eigenschaften. Diese Individuen werden durch die natürliche Auslese (**Selektion**) ausgewählt und haben bessere Fortpflanzungschancen, wodurch sie mehr Nachkommen haben und sich die besseren Eigenschaften ausbreiten. Die Variationen können einerseits durch zufällige Veränderungen der Gene hervorgerufen werden (**Mutation**) und andererseits durch den Austausch von Teilchromosomen entstehen (**Rekombination**). Die Evolution ist ein Kreislauf von Neuerschaffung \Rightarrow Überlebensprüfung \Rightarrow Vermehrung der Besten \Rightarrow Verbesserung des Vorhandenen. Dieser Kreislauf wird auf die evolutionären Algorithmen übertragen. Für die Erläuterung weiterer wichtiger Grundbegriffe der biologischen Evolution siehe z.B. [KBK⁺11].

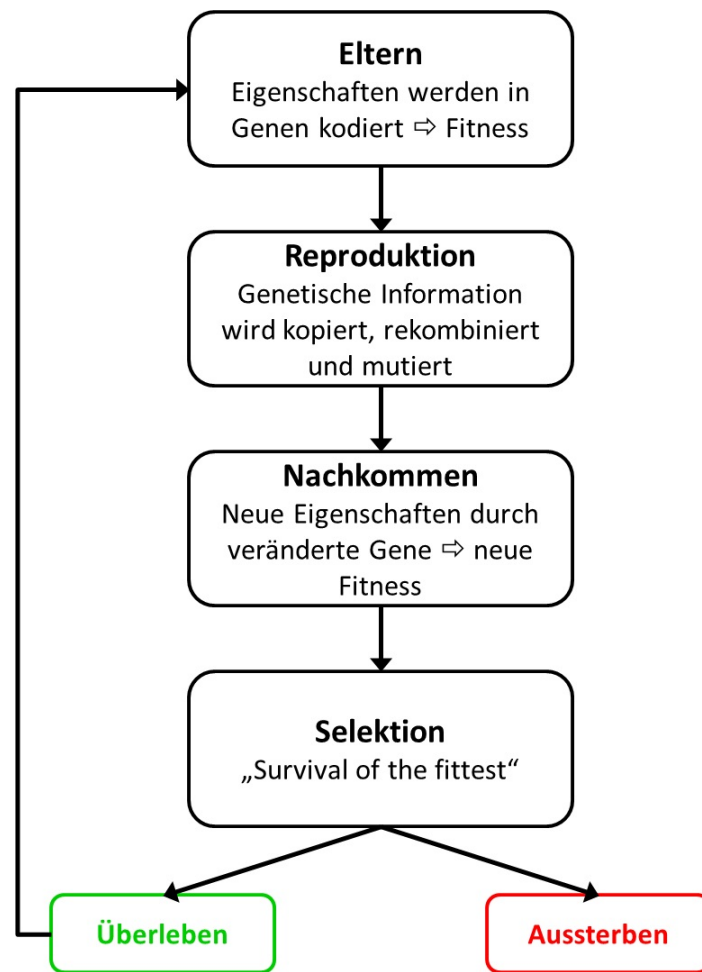


Abbildung 3.1: Genereller Ablauf eines evolutionären Algorithmus (vgl. [Bey01])

Die evolutionären Faktoren Mutation, Rekombination und Selektion werden in einen Algorithmus übersetzt. Dazu wird zunächst eine Startpopulation mit μ Individuen generiert. Jedes Individuum stellt hierbei eine potenzielle Lösung des Optimierungsproblems dar. Mithilfe einer Fitnessfunktion wird die Qualität jeder potenziellen Lösung bewertet. Anschließend werden in jeder Iteration die Lösungskandidaten in drei Schritten verbessert:

1. **Rekombination:** Es werden für jede Lösung ρ Eltern ausgewählt, meist $\rho = 2$, und deren Merkmale werden zu einer neuen Lösung kombiniert.
2. **Mutation:** Die Lösungen werden einer kleinen, zufälligen Änderung unterzogen.
3. **Selektion:** Auf Basis der Fitnesswerte werden von den erzeugten λ Nachkommen μ ausgewählt.

Diese Schritte werden solange durchgeführt bis das Abbruchkriterium erfüllt ist.

Bei der Evolutionsstrategie repräsentiert jedes Individuum einen Parametersatz \mathbf{z} des MOPs. Die

Initialisierung der Population ist beliebig und sie wird zunehmend verbessert durch die (zufallsbehafteten) Prozesse Selektion, Rekombination und Mutation. Der Selektionsprozess bevorzugt Individuen mit einem hohen Fitnesswert sich öfter zu reproduzieren als solche mit einer niedrigen Fitness. Der Fitnesswert ist hierbei identisch mit dem Funktionswert der Zielfunktion $Q(\mathbf{z}, t_f)$. Basierend auf dieser Information, verwendet die Evolutionsstrategie das Prinzip von Darwin „Survival of the fittest“. Der Rekombinationsprozess kombiniert zwei oder mehr Individuen zu neuen Individuen und bei der Mutation werden Individuen zufällig verändert, um die Population zu erneuern.

Folgende Bezeichnungen werden in diesem Abschnitt verwendet:

- $Q : Z \rightarrow \mathbb{R}$ Zielfunktion
- I Menge der Individuen
- $\phi : I \rightarrow \mathbb{R}$ Fitnessfunktion
- $a \in I$ ein Individuum
- $\mathbf{z} \in Z \subseteq \mathbb{R}^n$ Parametervektor
- $\mu \geq 1$ Größe der Population der Eltern,
- $\lambda \geq 1$ Größe der Population der Nachkommen, d.h. die Anzahl der Individuen, die durch Rekombination und Mutation in jeder Iteration erzeugt werden,
- $A(g) = \{a^1(g), a^2(g), \dots, a^\mu(g)\}$ Population in Generation g mit den Individuen $a^i(g) \in I$,
- $r_{\phi_r} : I^\mu \rightarrow I^\lambda$ Rekombinationsoperator, der die Nachkommen der Population produziert,
- $m_{\phi_m} : I^\lambda \rightarrow I^\lambda$ Mutationsoperator, der die Nachkommen der Population modifiziert,
- $s_{\phi_s} : W \rightarrow I^\mu$ Selektionsoperator, der die Eltern der nächsten Generation auswählt, mit $W = I^\lambda$ oder $W = I^{(\mu+\lambda)}$ und
- $\iota : I^\mu \rightarrow \{\text{wahr}, \text{falsch}\}$ Abbruchkriterium.

Nachfolgend wird die Evolutionsstrategie mithilfe dieser Bezeichnungen algorithmisch beschrieben.

Setze $g = 0$

Initialisiere $A(0) = \{a^1(0), a^2(0), \dots, a^\mu(0)\} \in I^\mu$

Berechne $A(0) : \{\phi(a^1(0)), \phi(a^2(0)), \dots, \phi(a^\mu(0))\}$

while not $\iota(A(g))$ **do**

Rekombiniere: $A''(g) = r_{\phi_r}(A(g))$

Mutiere: $A''(g) = m_{\phi_m}(A'(g))$

Berechne $A''(g) : \{\phi(a^1(g)), \phi(a^2(g)), \dots, \phi(a^\mu(g))\}$
 Selektiere: $A(g+1) = s_{\phi_s}(A''(g) \cup W)$, where $W \in \{\emptyset, A(g)\}$
 Setze $g = g + 1$

end while

Im Folgenden werden die Berechnung der Fitness, die Darstellung der Individuen und die drei Hauptprozesse – Rekombination, Mutation und Selektion – näher betrachtet.

3.2.1 Fitnessberechnung und Individuendarstellung

Der Fitnesswert eines Individuums ist identisch mit dem Zielfunktionswert des zugehörigen Parametersatzes, d.h. $\phi(\mathbf{a}^k) = Q(\mathbf{z}^k)$, wobei \mathbf{z}^k der Parametersatz ist, der zu dem Individuum \mathbf{a}^k gehört. Ein Individuum \mathbf{a}^k besteht aus einem Satz an Zielparametern \mathbf{z}^k und einem Satz an Strategieparametern \mathbf{s}^k

$$\mathbf{a}^k = (\mathbf{z}^k, \mathbf{s}^k) \in I.$$

Die Strategieparameter werden genutzt um den Mutationsprozess zu kontrollieren (siehe Abschnitt Mutation).

3.2.2 Rekombination

Bei der Rekombination wird ein neues Individuum aus zwei zufällig ausgewählten Elternindividuen erstellt. In der globalen Form ist es sogar erlaubt Komponenten für ein neues Individuum von potentiell allen Individuen der Elternpopulation zu kombinieren. Die Rekombination wird sowohl auf die Zielparameter als auch auf die Strategieparameter angewendet. Unter Umständen werden verschiedene Rekombinationsmechanismen für Ziel- und Strategieparameter genutzt:

$$a'_i = \begin{cases} a_i^S & \text{keine R.} \\ a_i^S \text{ oder } a_i^T & \text{diskrete R.} \\ a_i^S + \chi \cdot (a_i^T - a_i^S) & \text{generalisierte intermediäre R.} \\ a_i^{S_i} \text{ oder } a_i^{T_i} & \text{globale, diskrete R.} \\ a_i^{S_i} + \chi_i \cdot (a_i^{T_i} - a_i^{S_i}) & \text{globale, generalisierte intermediäre R.} \end{cases} \quad (3.1)$$

Die Indizes S und T bezeichnen zwei zufällig aus der Population $A(g)$ ausgewählte Elternindividuen und χ ist ein aus dem Intervall $[0; 1]$ beliebig wählbarer Parameter. Bei der globalen Rekombination werden die Eltern für jede Komponente z_i neu zufällig ausgewählt. Wird bei der (globalen) generalisierten intermediären Rekombination $\chi = 1/2$ bzw. $\chi_i = 1/2, \forall i \in \{1, 2, \dots, n\}$ gewählt, spricht man kurz von einer intermediären Rekombination.

Experimentelle Studien haben gezeigt, dass die diskrete Rekombination der Zielparameter und zusätzlich die intermediäre Rekombination der Strategieparameter die besten Ergebnisse erzielen [BS93].

3.2.3 Mutation

Bei der Mutation werden die Nachkommen, die durch die Rekombination generiert wurden, modifiziert. Hierbei können verschiedene Methoden angewendet werden, die alle eine bestimmte Anzahl an Strategieparameter benötigen. Die einfachste Mutationsmethode kommt mit einem Strategieparameter σ für die Mutationsstärke aus. Die Zielparameter werden dann wie folgt modifiziert

$$\begin{aligned} z' &= z + r \\ r &= \sigma \cdot N, \quad N = (N_1(0, 1), N_2(0, 1), \dots, N_n(0, 1)), \end{aligned}$$

wobei $N_i(0, 1)$ eine normalverteilte Zufallsvariable ist. Diese Methode kann erweitert werden indem jeder Parameter einen eigenen Strategieparameter für die Mutationsstärke zu gewiesen bekommt

$$r = (\sigma_1 N_1(0, 1), \sigma_2 N_2(0, 1), \dots, \sigma_n N_n(0, 1)).$$

Zusätzlich könnten auch noch die Korrelationen zwischen den Zielparametern berücksichtigt werden

$$r = M \cdot (\sigma_1 N_1(0, 1), \sigma_2 N_2(0, 1), \dots, \sigma_n N_n(0, 1))^T,$$

wobei M eine Rotationsmatrix ist, die die Korrelationen zwischen den einzelnen Zielparametern repräsentiert. Bei diesem Mutationsmechanismus werden $n(n+1)/2$ Strategieparameter verwendet.

Alle der aufgeführten Mutationsmechanismen benötigen eine angemessene Anpassung der Strategieparameter. Diese Anpassung kann durch die von Schwefel 1987 vorgestellte Methode der **Selbstadaption** erreicht werden [Sch87]. Dazu werden die Strategieparameter auch dem Mutationsprozess unterzogen und eventuell auch dem Rekombinationsprozess. Die mutierten Strategieparameter werden dann dazu genutzt den Mutationsprozess der Zielparameter zu kontrollieren.

Ein Strategieparameter wird wie folgt mutiert

$$\sigma' = \sigma \cdot \exp(\tau \cdot N(0, 1)),$$

wobei τ als Lernparameter bezeichnet wird und den Grad der Selbstadaption angibt. Theoretische und experimentelle Ergebnisse [Bey95],[Sch75] schlagen folgende Wahl des Lernparameters vor

$$\tau \propto \frac{1}{\sqrt{n}}.$$

Diese Technik kann entsprechend erweitert werden für den Fall, dass jeder Zielparameter seinen eigenen Strategieparameter hat

$$\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1)), \quad i = 1, 2, \dots, n. \quad (3.2)$$

Hierbei wird eine generelle Mutation mit einer Mutation für jede Komponente kombiniert. Die folgenden Werte werden für die Lernparameter τ' und τ empfohlen [Sch77]

$$\tau' \propto \frac{1}{\sqrt{2n}} \quad \tau \propto \frac{1}{\sqrt{2\sqrt{n}}}.$$

Für den Fall das auch Korrelationen betrachtet werden, müssen $n(n+1)/2$ Elemente der Matrix M modifiziert werden

$$m'_j = m_j + \beta \cdot N_j(0, 1), \quad j = 1, 2, \dots, n(n-1)/2,$$

wobei $\beta \approx 0.0873$ [Sch77]. Die Mutation der Standardabweichungen kann dann wie in Formel (3.2) durchgeführt werden.

Die Rekombination der Strategieparameter kann mit den Methoden in Formel (3.1) durchgeführt werden, wobei die intermediäre Rekombination empfohlen wird [BS02].

3.2.4 Selektion

Der Selektionsprozess ist ein komplett deterministischer Prozess mit zwei möglichen Strategien [BS93]:

- (μ, λ) -Selektion (**Komma-Selektion**): die μ besten Individuen der Menge der λ Nachkommen werden ausgewählt.
- (μ, λ) -Selektion (**Plus-Selektion**): die μ besten Individuen der Vereinigungsmenge der μ Eltern und λ Nachkommen werden ausgewählt.

Der Nachteil der Plus-Strategie ist die Unfähigkeit auf veränderte Umweltsituationen zu reagieren. Zusätzlich wird die Selbstadaption der Strategieparameter behindert, da falschangepasste Strategieparameter mehrere Generationen überleben können. Die Fähigkeit der Komma-Selektion gute Lösungen zu vergessen erlaubt ein Verlassen von lokalen Maxima und ist somit vorteilhaft für multimodale Zielfunktionen [Bäc96]. Die Komma-Selektion wird heutzutage empfohlen und Experimente zeigen, dass das optimale Verhältnis von Eltern und Nachkommen $\mu/\lambda = 1/7$ ist [Sch87].

3.3 Hybride Optimierungsverfahren

Der Nachteil der Evolutionsstrategie ist die hohe Rechenzeit aufgrund der langsamen Konvergenzraten. Sie arbeiten generell gut bei der Erkundung des Parameterraums, aber sie sind langsam beim genauen Auffinden des Maximums der Zielfunktion. Lokale Methoden sind um einiges schneller im Auffinden eines Maximums, wenn sie sich erstmal in der „richtigen“ Region des Parameterraums befinden. Andererseits können sie aber in einem lokalen Minimum stecken bleiben. Der Optimierungsprozess könnte entsprechend optimiert werden, wenn beide globale und lokale Methoden zu sogenannten **hybriden Methoden** kombiniert werden. Der Vorteil der hybriden Methoden ist die Erhöhung der Konvergenzgeschwindigkeit, wobei gleichzeitig das Steckbleiben in lokalen Maxima verhindert wird. Mit der Evolutionsstrategie wird eine erfolgversprechende Region im Parameterraum gesucht und die lokale Methode findet die beste Lösung in dieser Region schnell und genau. Globale und lokale Methoden können auf unterschiedliche Art kombiniert werden. Eine Übersicht dazu kann in [YLLR98] gefunden werden. Zwei Möglichkeiten sind in Abbildung 3.2 dargestellt. Bei beiden wird zunächst mithilfe der Evolutionsstrategie eine erfolgversprechende Region im Parameterraum gesucht. Ist diese gefunden wird der Parametersatz des besten Individuums aus der dieser Region als Startwert für die lokale Methode verwendet. Dieser Prozess könnte, wie in Abbildung 3.2 rechts dargestellt, mehrmals durchlaufen werden.

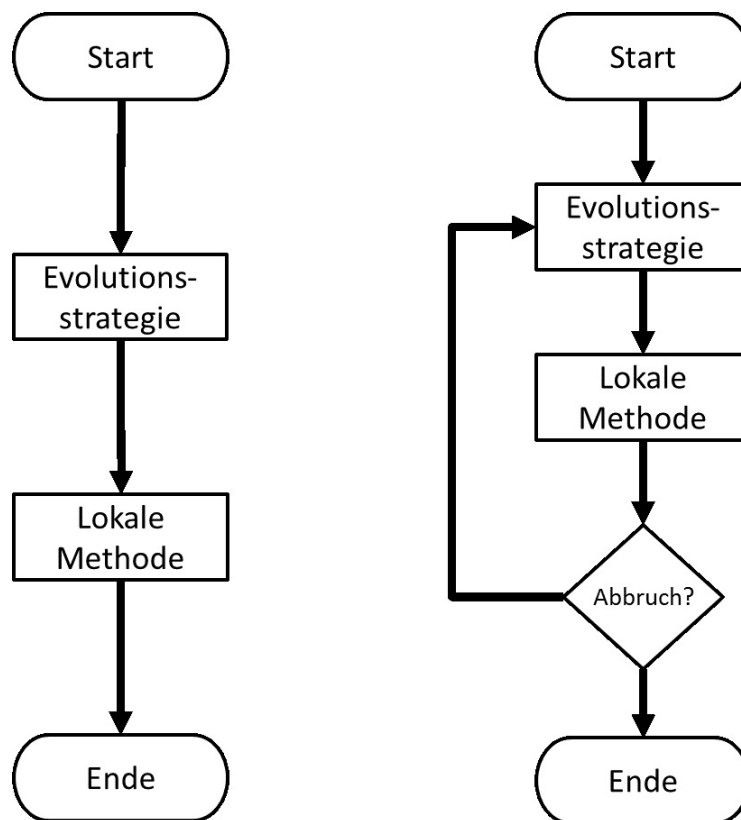


Abbildung 3.2: Ablauf von hybriden Optimierungsmethoden

4 Steuerung, Regelung und Automatisierung des Produktionsprozesses

Mit den vorgestellten Algorithmen lässt sich der optimale Parametersatz zur Steuerung eines Bioreaktors bestimmen. Wenn man das eingangs vorgestellte Beispiel der Xanthanproduktion betrachtet, könnte man beispielsweise die Parameter Stickstoffkonzentration (N_0), Glukosekonzentration (S_0) und Temperatur (T_0) variieren, da diese Parameter einen großen Einfluss auf das Wachstum und die Xanthanproduktion der Bakterien haben. Zudem könnte man zu einem Zeitpunkt (t_z) weitere Nährstoffe (N_z, S_z) hinzufügen. Auch die Temperatur (T_h) könnte zu einem Zeitpunkt (t_h) verändert werden. Der Zeitpunkt, zu dem die Xanthanausbeute betrachtet wird, kann entweder fest gewählt werden ($t_f = 100 [h]$) oder mit optimiert werden. Das MOP aus Abschnitt 2 kann dann wie folgt konkretisiert werden, wobei $P(z, t_f)$ die Xanthanausbeute zum Zeitpunkt t_f ist, die mit dem Parametersatz \mathbf{z} erreicht wurde:

$$Q_1 = P(\mathbf{z}, 100) \rightarrow \max$$

$$\text{oder } Q_2 = \frac{P(\mathbf{z}, t_f)}{t_f} \rightarrow \max$$

$$\text{mit } \mathbf{z} = (N_0, S_0, T_0, t_z, N_z, S_z, T_h, t_h).$$

Abbildung 4.1 zeigt Simulationsergebnisse von Glukose, Stickstoff, Xanthan und Biomasse¹ bei optimaler Steuerung des Bioreaktors. Als Zielfunktion wurde Q_1 verwendet und die oben genannten Parameter wurden bei der Optimierung mittels einem hybriden Optimierungsalgorithmus variiert. Man erkennt, dass bei der optimalen Steuerung der Fermentationsprozess in zwei Phasen eingeteilt wird. Die erste Phase ist die **Wachstumsphase** mit einer hohen Stickstoffkonzentration und einer niedrigen Glukosekonzentration. Aufgrund der niedrigen Glukosekonzentration wird in dieser Phase kaum Xanthan produziert und die Bakterien vermehren sich hauptsächlich. Die zweite Phase ist die **Xanthanproduktionsphase** und wird durch eine Hinzugabe von Glukose eingeleitet. Es wird kein weiterer Stickstoff hinzugefügt, so dass die Bakterien das Wachstum einstellen und sich stattdessen auf die Produktion von Xanthan konzentrieren. Zudem findet zum gleichen Zeitpunkt eine Erhöhung der Temperatur im Bioreaktor statt, um die Ausbeute an Xanthan noch zusätzlich zu erhöhen.

¹Masse der durch Lebewesen anfallenden organischen Substanz in einem bestimmten Lebensraum. Die Biomasse kann als Indikator für das Wachstum der Bakterien herangezogen werden.

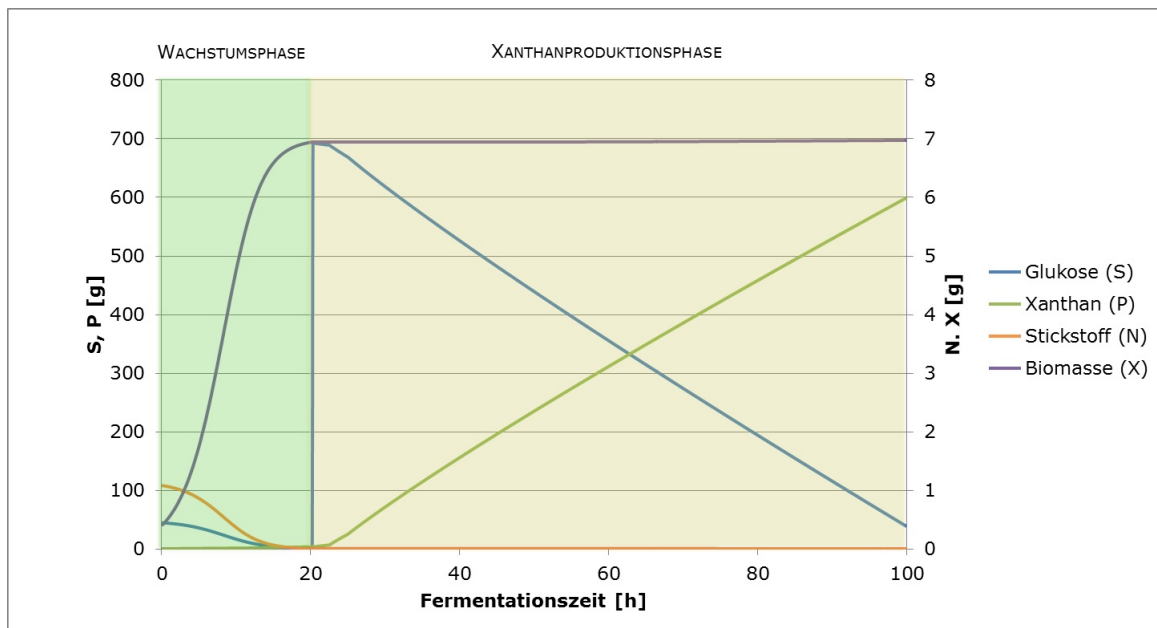


Abbildung 4.1: Glukose, Stickstoff, Xanthan und Biomasse bei optimaler Steuerung des Bioreaktors

Bei der oben genannten **Steuerung** eines Bioprozesses findet kein Abgleich von Ausgangs- und Eingangsgröße statt. Es handelt sich hierbei um einen offenen Wirkungsablauf, bei dem es keinerlei Rückkopplung gibt (siehe Abbildung 4.2). Somit ist es möglich, dass sich der Bioprozess aufgrund von Störungen ändert. Um dem entgegenzuwirken, könnte man eine **Regelung** für den betrachteten Bioprozess etablieren. Bei der Regelung wird eine vorgegebene Größe auf einen bestimmten Wert gebracht und dort gehalten oder die vorgegebene Größe wird an eine sich ändernde Größe angepasst.

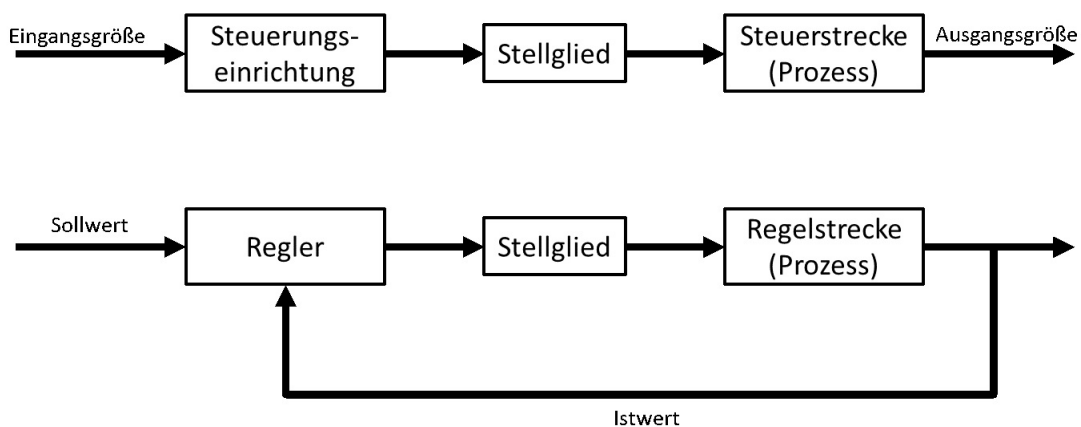


Abbildung 4.2: Steuerung (oben) und Regelung (unten) eines Prozesses

Ein weiterer immer bedeutungsvoller werdender Aspekt ist die **Automatisierung** des Bioprozesses. Diese Automatisierung könnte Kosten einsparen und die Produktausbeute erhöhen.

Abbildung 4.3 zeigt ein Konzept einer vollautomatisierten Regelung zur Maximierung der Produktausbeute. Die Regelung eines Prozesses aufgrund eines Modells wird in diesem Zusammenhang auch

modellprädikative Regelung, meist auch **Model Predictive Control (MPC)** genannt. Hierbei wird das Modell dazu genutzt in Abhängigkeit von aktuellen Messungen den Prozessstatus des realen Systems in der Zukunft zu berechnen. Basierend auf diesen Vorhersagen und einem Expertensystem im Hintergrund, können die Prozessparameter entsprechend angepasst werden. Mit zusätzlichen Methoden der künstlichen Intelligenz kann ein selbstlernendes System aufgebaut werden. Der Fermentationsprozess könnte hierbei vollautomatisiert ablaufen. Die Prozessdaten werden durch spezifische Sensoren gemessen und die Prozessparameter werden in Abhängigkeit der Kontrolldaten durch Aktoren entsprechend angepasst. Die Kontrolldaten werden durch mathematische Methoden und durch das Prozessmodell gewonnen.

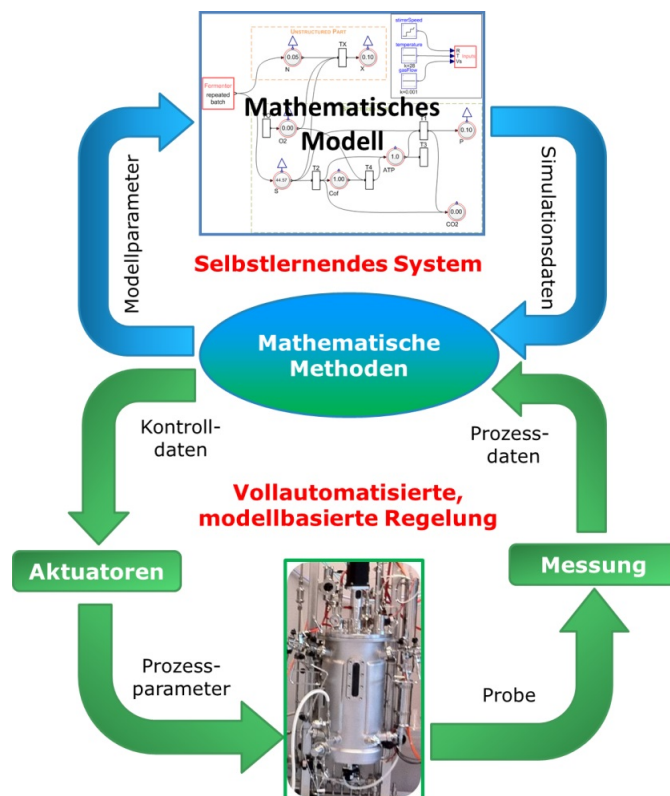


Abbildung 4.3: Vollautomatisierte Regelung zur Maximierung der Produktausbeute

5 Fazit

In diesem Artikel wurde eine allgemeine Vorgehensweise zur Maximierung der Produktausbeute bei biotechnologischen Produktionsprozessen vorgestellt. Zwei Beispiele, die Xanthanproduktion des Bakteriums *Xanthomonas campestris* und die Antikörperproduktion von CHO-Zellen, wurden zur Verdeutlichung der Problematik herangezogen.

Zunächst wurde aufgezeigt, wie basierend auf einem mathematischen Modell des Organismus und des Bioreaktors, das zugehörige mathematische Optimierungsproblem aufgestellt wird. Auf die Modellbildung selbst wurde hierbei nicht eingegangen.

Aufgrund der unerwünschten Eigenschaften der Zielfunktion in Kombination mit dem mathematischen Modell: nichtlinear, nicht stetig und nicht differenzierbar, eignen sich nur ableitungsfreie Methoden zur Lösung des Problems. Es wurde als lokale Methode der Nelder-Mead-Simplex-Algorithmus vorgestellt und als globale die Evolutionsstrategie. Zudem wurde aufgezeigt, wie die positiven Eigenschaften dieser beiden Methoden zu einer hybriden Lösungsstrategie kombiniert werden können.

Auf Basis der Ergebnisse dieser Optimierung wurde ein allgemeines Konzept zur Steuerung des Bioprozesses erläutert. Um auch auf Störungen während des Produktionsprozesses reagieren zu können, wurde eine modellbasierte Regelung präsentiert, die auch, um Kosten zu sparen, vollautomatisiert umgesetzt werden kann.

Literaturverzeichnis

- [Bäc96] BÄCK, T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, USA, 1996.
- [Bey95] BEYER, H.: Toward a theory of evolution strategies: On the benefits of sex. *Evolutionary computation*, 3(1):81–111, 1995.
- [Bey01] BEYER, H.: The theory of evolution strategies. Springer Verlag, Berlin, 2001.
- [BR06] BIRCH, JR. und RACHER, A.J.: Antibody production. *Advanced drug delivery reviews*, 58(5-6):671–685, 2006.
- [BS93] BÄCK, T. und SCHWEFEL, H.: An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23, 1993.
- [BS02] BEYER, H. und SCHWEFEL, H.: Evolution strategies—A comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.
- [Chm11] CHMIEL, H.: Bioprozesstechnik. Spektrum Akademischer Verlag, Heidelberg, 2011.
- [GSCG00] GARCIA-OCHOA, F., SANTOS, V., CASAS, JA. und GOMEZ, E.: Xanthan gum: production, recovery, and properties. *Biotechnology Advances*, 18(7):549–579, 2000.
- [KBK⁺11] KRUSE, R., BORGELT, C., KLAWONN, F., MOEWES, C., RUSS, G. und STEINBRECHER, M.: Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze. Vieweg+Teubner Verlag, 2011.

- [LRWW99] LAGARIAS, J., REEDS, J., WRIGHT, M. und WRIGHT, P.: Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM Journal on Optimization, 9(1):112–147, 1999.
- [LT97] LAO, MS. und TOTH, D.: Effects of ammonium and lactate on growth and metabolism of a recombinant Chinese hamster ovary cell culture. Biotechnology progress, 13(5):688–691, 1997.
- [NM65] NELDER, J. und MEAD, R.: A simplex method for function minimization. The computer journal, 7(4):308–313, 1965.
- [PBAS06] PROVOST, A., BASTIN, G., AGATHOS, SN. und SCHNEIDER, YJ.: Metabolic design of macroscopic bioreaction models: application to Chinese hamster ovary cells. Bioprocess and biosystems engineering, 29(5):349–366, 2006.
- [Pro13] PROSS, S.: Hybrid Modeling and Optimization of Biological Processes. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik, 2013. (ISSN 2196-6192).
- [Rec71] RECHENBERG, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Doktorarbeit, 1971.
- [Sch75] SCHWEFEL, H.: Evolutionsstrategie und numerische Optimierung. Doktorarbeit, 1975.
- [Sch77] SCHWEFEL, H.: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Birkhäuser Verlag, Basel Stuttgart, 1977.
- [Sch87] SCHWEFEL, H.: Collective phenomena in evolutionary systems. In: Preprints of the 31st Annual Meeting of the International Society for General System Research, Band 2, Seiten 1025–1033, 1987.
- [Wie02] WIECHERT, W.: Modeling and simulation: tools for metabolic engineering. Journal of biotechnology, 94(1):37–63, 2002.
- [YLLR98] YEN, J., LIAO, J., LEE, B. und RANDOLPH, D.: A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 28(2):173–191, 1998.

Simulation und Optimierung der Personalplanung im Pflegebereich von Krankenhäusern durch Petri-Netz-Modelle

Timo Lask¹, Hermann-Josef Kruse¹, Bernhard Bachmann¹

¹Fachbereich Ingenieurwissenschaften und Mathematik, FH Bielefeld

Kontakt:

timo.lask@fh-bielefeld.de

Zusammenfassung. Eines der Hauptprobleme aller Pflegeeinrichtungen ist (und wird auch in Zukunft sein) der Mangel an qualifizierten Pflegefachkräften. Als eine Hauptursache für die abnehmenden Bewerbungszahlen von Auszubildenden und der frühzeitige Berufsausstieg sind die Arbeitszeitregelungen im Schichtbetrieb und ihre Folgeerscheinungen. Aus diesen Gründen ist es für eine Pflegeeinrichtung wichtig, Arbeitszeitkonzepte zu entwerfen und anzubieten, die das Pflegepersonal möglichst gering belastet, um hierdurch Fachpersonal zu gewinnen bzw. zu halten.

In diesem Artikel wird gezeigt, wie mit Hilfe von Simulationsmodellen auf der Basis von Petri-Netzen verschiedene Arbeitszeitkonzepte vor ihrer Umsetzung auf Machbarkeit im Hinblick auf relevante Kriterien und Vorgaben zum Zwecke der Entscheidungsunterstützung überprüft werden können.

Schlüsselwörter. Petri-Netze, Multiagentensimulation, Pflege-Personalplanung, Arbeitszeitkonzepte

1 Problemstellung und Lösungsidee

Eines der Hauptprobleme aller Pflegeeinrichtungen ist der Fachkräftemangel. Zum einen sind der frühzeitige Berufsausstieg von Pflegepersonal nach durchschnittlich 14 Jahren [Hac09] und das durchschnittliche Rentenzugangsalter von 58,5 [Hie09] zu nennen. Zudem geben einige Krankenpflegeschulen an, dass die Bewerberzahlen in den letzten Jahren deutlich gesunken sind. Dies ist insgesamt eine drastische Entwicklung.

Die Ursachen für diese Probleme in der Krankenhauspflege sind vielfältig. Unter anderem aber ist eines der Hauptursachen die Arbeitszeitregelung im Schichtbetrieb und ihre Folgen. Die NEXT-Studie¹ hat nachgewiesen, dass die Zufriedenheit bzw. Unzufriedenheit von Pflegepersonal vor allem durch die Organisation der Arbeitszeit beeinflusst wird. Unter anderem empfinden 30% der Befragten die vorhandene Arbeitszeitregelung als belastend, gar 45% sehen ihre Work-Life-Balance gefährdet. Des Weiteren sehen jüngere Berufsangehörige und Einsteiger eine Unvereinbarkeit von Familie und Beruf. Insgesamt entstehen durch die Arbeitszeitregelung große soziale Defizite für die Arbeitnehmer(innen).

Aus oben genannten Gründen ist es für eine Pflegeeinrichtung wichtig, eine Arbeitszeitregelung zu entwerfen, die das Pflegepersonal möglichst gering belastet, um hierdurch Personal zu gewinnen bzw. zu halten. Dies muss allerdings unter der Bedingung geschehen, dass Patienten an 24 Stunden 365 Tage im Jahr hinreichend versorgt werden müssen und bestimmte Kosten nicht überschritten werden dürfen.

Beispielsweise arbeitet das Pflegepersonal in der geriatrischen Abteilung eines Krankenhauses² momentan in einer 6-Tage-Woche³ mit drei Schichten (Früh-, Spät-, Nachtschicht). Dies soll in ein System mit einer 5-Tage-Woche⁴, welche drei bzw. vier Schichten (Früh-, Tag-, Spät-, Nachtschicht) beinhaltet, geändert werden. Dies hätte zwei Effekte. Einerseits würde sich die Anzahl der potenziellen freien Tage verdoppeln, was mehr Raum für soziale Aktivitäten bieten würde. Der andere Effekt wäre, dass die Arbeitszufriedenheit durch eventuell vier Schichten steigen würde, da nach [HM05] die Arbeitszufriedenheit bei Pflegenden im Drei-Schichten-System am wenigsten ausgeprägt ist.

¹Die NEXT-Studie (nurses' early exit study) hat Ursachen, Umstände und Folgen des vorzeitigen Berufsausstiegs aus der Pflege untersucht. Online verfügbar: <http://www.next.uni-wuppertal.de> [Stand 24.10.2014].

²Referenzstation eines Kooperationspartners im BMBF-Projekt *Flexible und intelligente Pflegepersonalplanung für ein demografie-festes Krankenhaus* (Filip), 2015-2018.

³Dies bedeutet, dass eine Vollzeitkraft in der Regel 12 Tage am Stück arbeitet und dann 2 Tage frei hat.

⁴Dies bedeutet, dass eine Vollzeitkraft in der Regel 10 Tage am Stück arbeitet und dann 4 Tage frei hat. Die vermeintlich längere (6-Tage) bzw. kürzere (5-Tage) Arbeitszeit wird durch eine Anpassung der Schichtzeiten ausgeglichen.

Dies sind gravierende Umstellungen, die das Krankenhaus plant. Daher stellen sich folgende Fragen: „Reicht das vorhandene Personal für die Umstellung aus? Wenn nein, unter welchen Bedingungen könnte es ausreichen?“. Die Beantwortung dieser Fragestellungen ist äußerst schwierig, da sehr viele Einflussgrößen und Nebeneffekte aufgrund komplexer Zusammenhänge bei der Umstellung des Arbeitszeitmodells zu beachten sind, so dass eine manuelle Planung wegen des damit verbundenen Zeitaufwandes nicht bzw. nur sehr schwer realisierbar wäre. Daher soll ein Simulationsmodell erstellt werden, mit dem a priori überprüft werden kann, ob das bestehende Personal für eine geplante bzw. ausgedachte Personaleinsatz-Umstellung ausreicht.

Hieraus ergibt sich eine Modellierungsproblematik für den Planer. Es sollte eine Modellierungsmethode gewählt werden, mit dem das komplexe Arbeitszeitverhalten im Pflegebereich abgebildet werden soll. Eine weitere Anforderung ist, dass auch eine für den Probleminhaber verständliche Modellierungsmethode gewählt werden soll. Hierzu wird eine **Multiagentensimulation (MAS)** wie bei der Problemstellung von [SGW12] verwendet. Bei einer MAS werden ein (oder mehrere) Agenten implementiert, welche Objekte (Einheiten, Akteure, Personen, usw.) eines realen Systems darstellen. Diese Agenten sollen das Verhalten (bzw. die Eigenschaften) ihrer „realen Gegenstücke“ im Modell nachahmen. Hierdurch soll ein „reales“ System hinreichend genau abgebildet werden, so dass man durch die MAS einen Erkenntnisgewinn über das System erhält bzw. mit der MAS „herumexperimentieren“ kann.

MAS zeichnen sich durch folgende Eigenschaften aus (vgl. [SGW12]):

- System aus gleichartigen oder unterschiedlichen spezialisierten Agenten, die kollektiv ein Problem lösen
- Besonders geeignet zur Modellierung komplexer Systeme
- Kein vordefiniertes Verhaltensskript, sondern Reaktion auf konkrete Situation.

Der Grundaufbau eines MAS wird in Abb. 1.1 dargestellt.

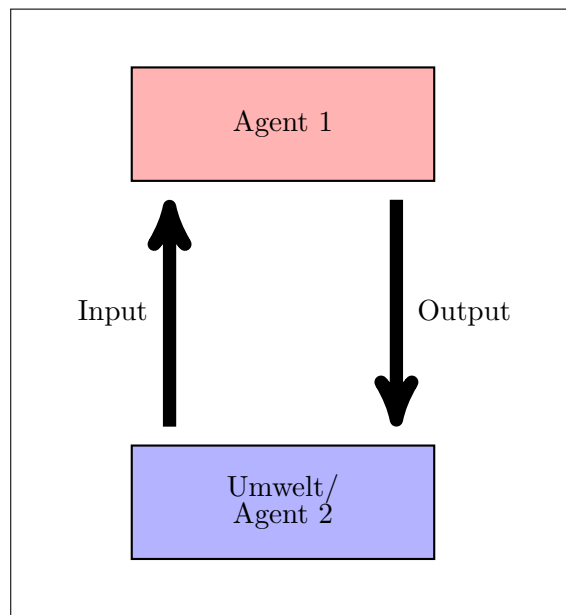


Abbildung 1.1: Grundaufbau eines MAS

Dieses Konzept der Simulation soll nun auf die Problemstellung der Pflegepersonalplanung angewendet werden.

Die einzelnen Agenten für die MAS werden hier mithilfe von Petri-Netz-Modellen beschrieben (vgl. [BKK⁺14] und [Pro13]). Die folgenden Agenten wurden implementiert:

- Station (Umwelt)
- Pflegekraft

Welche Agenten man in ein Modell aufnimmt, ist immer davon abhängig, welche Fragestellung untersucht werden soll. Mit diesen zwei Agenten wird zunächst eine hinreichend genaue Abbildung der Realität erreicht. Zur Verfeinerung des Modells könnte man weitere Agenten (Ärzte und Patienten) aufnehmen, um die Einflüsse der Schichtumstellung auf diese als indirekte Beteiligte zu untersuchen. Hierauf wird in diesem Grundmodell verzichtet.

Es werden mehrere Simulationsmodelle für unterschiedliche Arbeitszeitmodelle (6-Tage-Woche oder 5-Tage-Woche, drei bzw. vier Schichten, usw.) erstellt. Unter anderem ist auch ein Modell der 6-Tage-Woche und mit drei Schichten, welches den Ist-Zustand abbildet, erstellt worden. Dieses Modell soll zur Validation der in den Modellen zugrundegelegten Annahmen dienen und zum Vergleich mit neuen bzw. geplanten Arbeitszeitmodellen verwendet werden, um hieraus die Machbarkeit abschätzen zu können.

Alle Simulationsmodelle folgen dem in Abb. 1.2 dargestellten Aufbau.

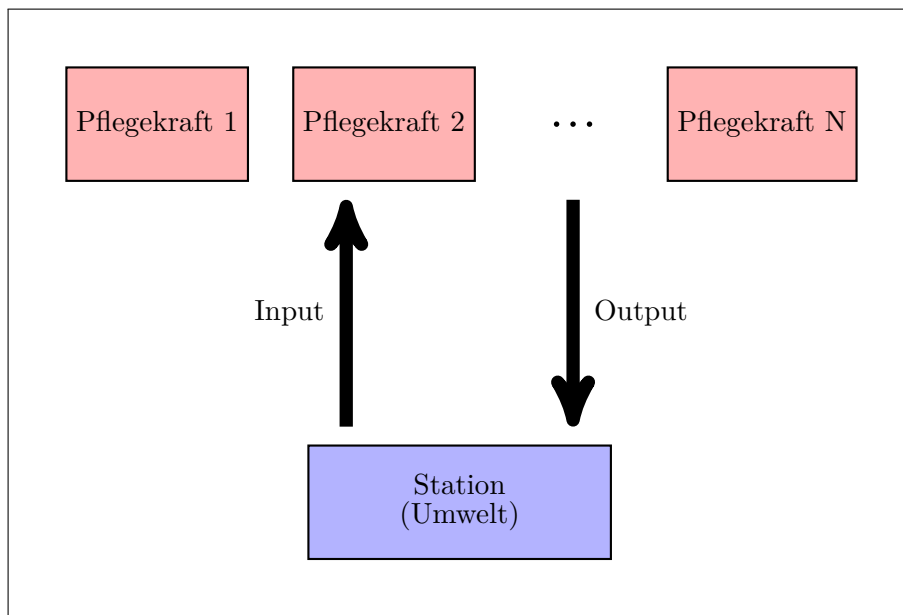


Abbildung 1.2: Aufbau der Simulationsmodelle

Zwecks guter Vergleichbarkeit mit der Realität wurde beschlossen, ein komplettes Jahr zu simulieren, weil hier alle Ereignisse des Jahres einmal durchlaufen werden. Es wurde der Zeitraum vom 1. August 2013 bis 31. Juli 2014 gewählt, weil dieser komplett in der Vergangenheit liegt und so alle nötigen Daten vorhanden sind. Start der Simulation ist am 1. August 2013 um 0:00 Uhr. Die erste simulierte Schicht ist die Frühschicht an diesem Tag, wobei die angebrochene Nachtschicht vernachlässigt wird. Die letzte berücksichtigte Schicht in der Simulation ist die Nachtschicht vom 31. Juli 2014 auf den 1. August 2014, so dass jede Schicht 365 mal durchlaufen wird.

Es sei angemerkt, dass zur einfachen Betrachtung eine Sekunde in der Modellimplementierung eine Stunde in der Realität darstellen soll. Einige wichtige Zeitwerte sind in der folgenden Tabelle aufgelistet:

Tabelle 1.1: Realzeit-Umrechnung in Modellzeit

Realzeitzeit	Modellzeit
1 Stunde	1 Sekunde
1 Tag	24 Sekunden
1 Woche	168 Sekunden
2 Wochen	336 Sekunden
4 Wochen	672 Sekunden
1 Jahr	8760 Sekunden

In den Agenten müssen viele Eigenschaften abgebildet werden. Diese einzelnen Spezifikationen, die im Modell berücksichtigt werden, sind im folgenden Abschnitt erläutert. An dieser Stelle sei noch angemerkt, dass es bei der hier behandelten Fragestellung nicht darum geht, konkrete Schichtpläne

zur Unterstützung der Personaleinsatzplanung für den laufenden Betrieb in einem Krankenhaus zu entwickeln, auch wenn viele Modellkomponenten bei den Lösungsansätzen für diese andere Zweckbetrachtung auf kanonische Weise ebenfalls vorkommen (hierzu siehe [Röm15], [BDBV04], [BC14]).

2 Modell-Spezifikationen

In den folgenden Abschnitten wird die Modellierung der einzelnen Eigenschaften (Problemstellung) im Modell erläutert. Für eine bessere Unterscheidbarkeit zwischen den Komponenten des Realsystems und den Modellkomponenten werden die realen Gegebenheiten in *kursiver Schrift* hervorgehoben.

Die im Simulationsmodell zu berücksichtigenden Nebenbedingungen aus der konkreten Praxisproblematik können für den „Probleminhaber“ unterschiedlich abgestufte Prioritäten hinsichtlich ihres Erfüllungsgrades haben. Im Folgenden wird der Einfachheit halber zwischen zwei „Bedingungsarten“ unterschieden: *harte* und *weiche* Bedingungen. Eine *harte Bedingung* ist auf jeden Fall exakt einzuhalten, sodass bereits eine marginale Überschreitung der gesetzten Grenzen zur Unzulässigkeit eines entsprechenden Lösungsvorschlages führt. Demgegenüber wird bei einer *weichen Bedingung* eine (in der Regel marginale) Grenzüberschreitung akzeptiert, auch wenn die Einhaltung der Bedingung grundsätzlich und nach Möglichkeit angestrebt wird; dabei sind die noch abnehmbaren Grenzüberschreitungen („bis maximal“ bzw. „in Einzelfällen“) vom Probleminhaber vorzugeben.

Die im Modell verankerten harten Bedingungen können in der „gelebten“ Personaleinsatzplanung durch Ausnahmereinbarungen flexibler geregelt werden. Dieser Unterschied muss selbstverständlich bei der Interpretation der Simulationsergebnisse berücksichtigt werden. Dies gilt ebenso für andere Modellvereinfachungen.

2.1 Verschiedene Berufsqualifikationen

Im Pflegebereich eines Krankenhauses arbeiten viele verschiedene Berufsqualifikationen mit einem jeweiligen Fähigkeiten- und Aufgaben-Spektrum, unter anderem: Krankenschwester/-pfleger, Altenpfleger/in, Pflegehelfer/in, Praktikant/in, Schüler/innen (1 Jahr, 2 Jahr, 3 Jahr), Arzthelfer/in. Die unterschiedlichen Berufsqualifikationen sind in der Praxis von großer Relevanz, da bestimmte Tätigkeiten nur von bestimmten Qualifikationen übernommen werden dürfen, z.B. darf nur eine Krankenschwester oder ein Krankenpfleger Erythrozyten-Konzentrat (EK) transportieren.

Für das Modell wurden die beruflichen Qualifikationen des Pflegepersonals in 4 Gruppen unterteilt:

- Pflegefachkraft (m/w)

Personalplanung im Pflegebereich

- Helfer (m/w)
- Praktikant (m/w)
- Schüler (m/w)

2.2 Arbeitszeitmodelle

Eine Pflegeeinrichtung muss 24 Stunden am Tag mit Pflegepersonal besetzt sein. Um dies zu gewährleisten, wird dort im Schichtbetrieb gearbeitet. Es existieren einige Grundkonzepte, z.B. drei bzw. vier Schichten, dazu gibt es meist noch weitere „spezielle“ Schichten wie später Frühdienst bzw. früher Spätdienst usw., welche um ein paar Stunden verschobene Schichten sind. Die Länge einer Schicht richtet sich in der Regel danach, wie viele Tage in der Woche das Pflegepersonal arbeiten soll; demnach sind Schichten in einer 5-Tage-Woche länger als Schichten in einer 6-Tage-Woche. Wann eine Schicht endet und die nächste beginnt, kann sich von Arbeitszeitmodell zu Arbeitszeitmodell unterscheiden, allerdings muss eine gewisse Überschneidungszeit der Schichten mit eingeplant werden, um eine Übergabe durchzuführen. Sollte auf einer Station viel zu tun sein, kann es passieren, dass das Pflegepersonal etwas länger arbeitet, als die eigentliche Schicht dauert (Überstunden). Auf der Referenzstation wird momentan mit einem Drei-Schichten-System mit teilweisen Spezialschichten gearbeitet. Aus arbeitsrechtlichen Gründen enthält jede Schicht eine Pause, deren Dauer sich nach der jeweiligen Schichtdauer richtet.

Im Modell werden die Schichtzeiten als harte Bedingungen abgebildet; demnach sind keine Überstunden möglich. Des Weiteren werden zur einfacheren Betrachtung Spezialschichten vernachlässigt. Für das Modell werden drei verschiedene Arbeitszeitkonzepte erstellt.

6-Tage-Woche mit drei Schichten

Die 6-Tage-Woche mit drei Schichten ist die „vereinfachte“ IST-Situation der Referenzstation; daher wurden hier die momentanen Schichtzeiten verwendet.

Tabelle 2.1: Zeiten bei 6-Tage-Woche mit drei Schichten

Schicht	Beginn	Ende	Arbeitszeit	Pause
Früh	6:00 Uhr	13:30 Uhr	7 Stunden	0,5 Stunden
Spät	13:00 Uhr	20:00 Uhr	7 Stunden	0,5 Stunden
Nacht	19:45 Uhr	6:15 Uhr	9,75 Stunden	0,75 Stunden

5-Tage-Woche mit drei Schichten

Für die 5-Tage-Woche mit drei Schichten wurden folgende geplante Schichtzeiten verwendet.

Tabelle 2.2: Zeiten bei 5-Tage-Woche mit drei Schichten

Schicht	Beginn	Ende	Arbeitszeit	Pause
Früh	6:00 Uhr	14:12 Uhr	7,7 Stunden	0,5 Stunden
Spät	12:18 Uhr	20:30 Uhr	7,7 Stunden	0,5 Stunden
Nacht	20:15 Uhr	6:15 Uhr	9,25 Stunden	0,75 Stunden

5-Tage-Woche mit vier Schichten

Für die 5-Tage-Woche mit vier Schichten wurden folgende geplante Schichtzeiten verwendet.

Tabelle 2.3: Zeiten bei 5-Tage-Woche mit vier Schichten

Schicht	Beginn	Ende	Arbeitszeit	Pause
Früh	6:00 Uhr	14:12 Uhr	7,7 Stunden	0,5 Stunden
Tag	7:30 Uhr	15:42 Uhr	7,7 Stunden	0,5 Stunden
Spät	12:18 Uhr	20:30 Uhr	7,7 Stunden	0,5 Stunden
Nacht	20:15 Uhr	6:15 Uhr	9,25 Stunden	0,75 Stunden

2.3 Sollbesetzung, Einsatzplanung und Pflegedefizit

Jede Station benötigt eine gewisse Sollbesetzung¹ in den einzelnen Schichten, um die anfallenden Aufgaben in einem „ausreichenden Maße“ zu erledigen und um Qualitätsstandards zu erfüllen. Diese Besetzung wird in der Regel über „Experten-Meinung“ bestimmt und ist daher kein Evidenz-basierter Wert. Für die Personaleinsatzplanung wird „einige Wochen“ im Voraus ein Dienstplan erstellt, in dem festgelegt wird, in welcher Schicht welche Pflegekräfte eingesetzt werden. Nachdem ein solcher Dienstplan erstellt wurde, kann es vorkommen, dass Pflegekräfte untereinander Schichten tauschen; dies kann auch sehr kurzfristig passieren (sprich: „Hey, können wir beide morgen Früh- und Spätschicht tauschen?“).

Bei der ganzen Personaleinsatzplanung wird darauf geachtet, die Sollbesetzung zu erreichen. Sollte aus Krankheits- oder anderen Gründen Personal kurzfristig ausfallen, wird versucht, Ersatz dafür zu finden, was bedeuten kann, dass Personal aus seinem freien Wochenende oder Ruhezeit geholt wird bzw. Überstunden macht, was nicht optimal für das jeweilige Personal ist. Ein Einsatz von mehr Personal als die Sollbesetzung kann auch angesetzt werden, damit das Pflegepersonal auf seine zu leistende Stundenanzahl kommt; dies wird aber nach Möglichkeit vermieden, weil es nicht „effizient“ genutzte Arbeitszeit ist. Des Weiteren könnten „höher“ qualifizierte Pflegekräfte auch den Bedarf von „niedriger“ qualifizierten Pflegekräfte decken, was aber aus wirtschaftlicher Sicht nicht sinnvoll ist.

¹Unterschiedliche Berufsqualifikationen müssen hierbei berücksichtigt werden.

Sollbesetzung

Die Sollbesetzung einer Station gibt an, wie viele Pflegekräfte mit jeweiliger Qualifikation in den verschiedenen Schichten bzw. Zeiten auf einer Station vorhanden sein sollen. In der unteren Tabelle wird die momentane Sollbesetzung für drei Schichten dargestellt.

Tabelle 2.4: Sollbesetzung bei drei Schichten

Schicht	Pflegefachkraft	Helfer	Praktikant	Schüler
Früh	5	2	2	1
Spät	4	2	1	2
Nacht	2	0	0	0

Da für ein Vier-Schichten-System (noch) keine genauen Erfahrungswerte vorliegen, sollten hier verschiedene Sollbesetzungen ausprobiert werden, um zu ermitteln, welche die effizienteste¹ ist. Zwei Beispiele sind in den unteren Tabellen dargestellt.

Tabelle 2.5: Sollbesetzung bei vier Schichten (1)

Schicht	Pflegefachkraft	Helfer	Praktikant	Schüler
Früh	2	1	1	1
Tag	3	2	1	1
Spät	2	1	1	1
Nacht	2	0	0	0

Tabelle 2.6: Sollbesetzung bei vier Schichten (2)

Schicht	Pflegefachkraft	Helfer	Praktikant	Schüler
Früh	2	1	1	1
Tag	4	2	1	1
Spät	2	1	1	1
Nacht	2	0	0	0

Personaleinsatzplanung

Im Modell werden die Schichten einzeln betrachtet in der Reihenfolge, wie sie geleistet werden müssen (vgl. Abb. 2.1).

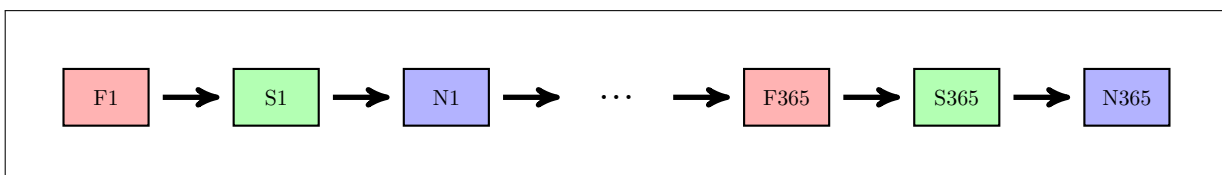


Abbildung 2.1: Simulationsablauf der Schichten

¹Hierbei muss auch darauf geachtet werden, ob die Ablaufprozesse auf der Station auf die angedachte Besetzung angepasst werden können.

Es wird immer versucht werden, die Sollbesetzung genau zu decken, daher also auch nicht mehr Personal einzusetzen als benötigt wird, um nicht durch ein zu frühes Verbrauchen von Arbeitszeitressourcen eine spätere Schicht nicht im Sollzustand besetzen zu können. Dadurch kann es in der Simulation dazu kommen, dass nicht das gesamte Personal seine Sollarbeitszeit erreicht. In der Realität würde dieses Personal dann einfach in einer Schicht, die schon den Sollbedarf erreicht hat, zusätzlich eingeplant werden.

Welches Personal (Pflegekraft A oder Pflegekraft B) konkret in persona eingesetzt werden soll, wird zufällig entschieden (gleichverteilt); hierbei werden nur die unterschiedlichen Berufsqualifikationen berücksichtigt. Im Modell ist die Möglichkeit, dass „höher“ qualifizierte Pflegekräfte den Bedarf von „niedriger“ qualifizierten Pflegekräften deckt, vorerst¹ ausgeschlossen, damit gezielt Aussagen darüber getroffen werden können, wie es um den Bedarf nach den jeweiligen Berufsqualifikationen steht.

Vergleichsgröße: Pflegedefizit

Für die Vergleichbarkeit von verschiedenen Arbeitszeitmodellen ist es wichtig, eine Zielgröße zu finden, mit der „einfach“ und dennoch „aussagekräftig“ ein Gütevergleich gemacht werden kann. Dazu wird das **Pflegedefizit** einer Schicht x wie folgt berechnet:

$$Pflegedefizit(x) := (Sollbesetzung(x) - Istbesetzung(x)) * Arbeitszeit(x)$$

Pflegedefizit [h]

Sollbesetzung [Mitarbeiter]

Istbesetzung [Mitarbeiter]

Arbeitszeit [h/Mitarbeiter]

Kann eine Schicht nicht mit dem Sollbedarf besetzt werden (aufgrund von Ruhezeit, Krankheit, Urlaub usw.), wird ein Pflegedefizit größer Null gemessen. Alle Pflegedefizite werden anschließend akkumuliert zu einem Pflegedefizit im Laufe des Jahres. Dies geschieht getrennt für die einzelnen Berufsqualifikationen, so dass später auch eine Aussage darüber möglich ist, in welcher Qualifikationsgruppe welches Jahresdefizit auftritt.

Das Pflegedefizit aus der Modellrechnung kann man wie folgt interpretieren: Je größer das Pflegedefizit eines Arbeitszeitmodells ist, desto öfter müssten in einem Realsystem entweder,

- Pflegepersonal eingesetzt werden, welches eigentlich nicht eingesetzt werden darf/sollte, da es sich im Zustand „Frei“ (wie Wochenende, Ruhezeit, Urlaub usw.) befindet.

oder

- die Schichten unterbesetzt absolviert werden.

¹Dies kann durch einen geringfügigen Aufwand in dem Modell umgesetzt werden.

Personalplanung im Pflegebereich

Beide Optionen führen zu mehr „Stress“ beim Pflegepersonal, beeinflussen aber auch die Qualität der Pflege an sich (Überarbeitung und Zeitmangel). Aus beiden Optionen würde daher auch wieder eine größere Arbeitsunzufriedenheit entstehen. Demnach ist ein Arbeitszeitmodell zu bevorzugen, welches ein möglichst niedriges Pflegedefizit verursacht.

2.4 Schüler (m/w)

Die meisten Pflegeeinrichtungen sind auch Ausbildungsbetriebe und haben daher auf den Stationen auch Schüler im Einsatz. Auch das St.Johannisstift bildet Pflegepersonal aus. Die Ausbildung einer Krankenschwester bzw. eines Krankenpflegers dauert 3 Jahre und die Ausbildung eines Pflegehelfers bzw. einer Pflegehelferin dauert 1 Jahr. Je nach Ausbildungsstand (1. Jahr, 2. Jahr oder 3. Jahr) können die Schüler unterschiedliche Aufgaben erledigen. Dabei wechseln sich Theorie- und Praxis-Phasen in mehrwöchigen Blöcken ab. Während der Ausbildung durchlaufen die Schüler mehrere Stationen, um verschiedene Fachrichtungen kennen zu lernen. Durch diese Umstände kann eine große Schwankung in der Anzahl der Schüler, die sich auf einer Station befinden, entstehen (teilweise befinden sich gar keine Schüler auf Station). Der Beginn eines Praxisblockes ist auch nicht unbedingt an Monatszyklen angepasst. Des Weiteren wird allen Schülern eines Kurses bis auf wenige Tage gleichzeitig Urlaub gegeben.

Da Schüler in nahezu allen Aspekten eine spezielle Modellierung benötigen würden, um sie hinreichend genau abzubilden, und das Modell nur „unnötig“ komplex machen würde für eine Personalgruppe, die im Schnitt nur 2 bis 3 Pflegekräfte pro Station darstellt, wurde die Gruppe der Schüler für das Modell außer Acht gelassen.

2.5 Stellenanteil im Zeitverlauf

Der Pflegebereich ist ein Sektor, in dem sehr viele Teilzeitkräfte eingesetzt werden. Bei der Referenzstation sind es 22 Teilzeitkräfte (TK) gegenüber 21 Vollzeitkräften (VK), wovon viele VK Praktikanten sind. Aufgrund von Zeitverträgen bzw. Vertragsänderungen und Arbeitsplatzwechsel besteht eine gewisse Fluktuation der Stellen(-anteile) auf einer Station. Im Schnitt bleibt der Stellenanteil allerdings gleich, weil für eine Vollzeit-Pflegekraft, die den Arbeitsplatz wechselt, eine neue „Äquivalente“ eingestellt werden muss. Dies dauert in der Regel allerdings einige Wochen, so dass die Fluktuation des Stellenanteils nicht vollständig verhindert werden kann.

Im Modell wird jeder Pflegekraft ihr jeweiliger Stellenanteil durch einen Parameter eingegeben. Dieser Stellenanteil ändert sich im Verlauf der Simulation nicht, sodass über die gesamte Simulationsdauer eine fest definierte Anzahl von Pflegekräften mit ihrem jeweiligen fixen Stellenanteil arbeiten. Dies hat vorrangig den Grund, das Modell (an dieser Stelle) nicht unnötig komplex zu machen. Es wurde das Personal (mit dem Stellenanteil) aus dem Juli 2014 verwendet.

2.6 Wöchentliche Arbeitszeit

Es gelten für Pflegekräfte einige rechtliche bzw. tarifliche Bedingungen für die wöchentliche Arbeitszeit, die eingehalten werden müssen:

- *38,5 Std./Woche im Durchschnitt von 26 Wochen*
- *Wöchentliche Arbeitszeit von 48 Std. (Ausgleich innerhalb von 26 Wochen)*
- *Wöchentliche Höchstarbeitszeit von 60/70 Std.*

Im Modell ist für jede Pflegekraft ein Arbeitszeitkontingent definiert. Eine Pflegekraft kann nur eingesetzt werden in einer Schicht, wenn ihr Arbeitszeitkontingent die Absolvierung der gesamten Schichtdauer zulässt (harte Bedingung). Im Modell wird im 14-Tage-Zyklus (Montag bis Sonntag) die Arbeitszeit geregelt, daher 77 Stunden innerhalb von 14 Tagen. Diese jeweilige zu leistende Arbeitszeit des Pflegepersonals wird nach dem jeweiligen Stellenumfang angepasst (sprich: eine Halbzeitstelle muss nur 38,5 Stunden arbeiten). Nach 14 Tagen wird die nicht verbrauchte Restarbeitszeit gelöscht (Arbeitszeitkontingent=0), im Anschluss werden dem Arbeitszeitkontingent 77 Stunden (bzw. ein der Stelle entsprechender Anteil) hinzugefügt. Hierdurch werden alle Bedingungen automatisch erfüllt, solange keine Schicht länger als 10 Arbeitsstunden dauert.

Des Weiteren wird den Pflegekräften ein Startkontingent an Arbeitszeit mitgegeben (angepasst an der Stelle und dem angebrochenen Arbeitszyklus); auf diese Weise lässt sich ein Startzustand erstellen.

2.7 Ruhezeiten

Aus arbeitsrechtlichen Gründen muss eine Pflegekraft, nachdem sie gearbeitet hat, eine Ruhezeit von 11 Stunden haben, bevor sie wieder eingesetzt werden darf. Es existieren gewisse Ausnahmen, wobei die Ruhezeit auf 10 Stunden reduziert werden darf. Die Ruhezeit sorgt in der Praxis dafür, dass eine Pflegekraft nicht von Spätdienst auf Frühdienst wechseln kann, ohne dass besondere Schichten wie früher Spätdienst bzw. später Frühdienst (mit einer eventuellen Reduzierung der Ruhezeit auf 10 Stunden) angewendet werden müssen. Ein schlechtes Ruhezeit-Management kann zu einer größeren Stressbelastung beim Pflegepersonal beitragen; deswegen sollten möglichst lange Ruhezeiten ermöglicht werden.

Die Ruhezeit wurde als harte Bedingung im Modell umgesetzt. Nachdem eine Pflegekraft eine Schicht absolviert hat, geht sie für exakt 11 Stunden in eine Ruhezeit und ist daher nicht dienstbereit, kann selbst „notfallmäßig“ nicht eingesetzt werden. Nach den 11 Stunden Ruhezeit kann die Pflegekraft wieder eingesetzt werden, dies muss allerdings nicht unmittelbar nach der Ruhezeit geschehen.

2.8 Wochenendregelung

Nach Tarifrecht gelten für Pflegekräfte einige Bedingungen für die Wochenendregelung:

- *mindestens jedes zweite Wochenende frei*
- *Sonntagsausgleichstag innerhalb von 14 Tagen*

Nach Möglichkeit wird immer versucht, dass die Tarifregelung eingehalten wird, aber auf Grundlage von „Personalmangel“ kann es passieren, dass die Wochenendregelung nicht eingehalten wird. Um eine 24-Stunden-Betreuung zu gewährleisten, wird das Pflegepersonal in zwei Gruppen aufgeteilt, die im Wechsel frei bekommen. In der Praxis kann es dazu kommen, dass Pflegepersonal kurzzeitig die Gruppe wechselt und an einem anderen Wochenende frei hat. Von dem Pflegepersonal wird ein „geblocktes Wochenende“ (alle freien Tage am Stück) meist bevorzugt. In „Notsituationen“ kann es passieren, dass eine Pflegekraft aus dem Wochenende geholt wird. Des Weiteren gibt es spezielle Verträge, in denen geregelt ist, dass keine Wochenendarbeit geleistet wird.

Für das Pflegepersonal wurden drei verschiedene Wochenendregelungen im Modell implementiert, die dann für die jeweilige Pflegekraft als Parameter in Anlehnung an die konkrete Istsituation der Referenzstation eingestellt werden. Folgende Möglichkeiten wurden implementiert:

- 1. Wochenende frei
- 2. Wochenende frei
- jedes Wochenende frei

Demnach ist das Pflegepersonal in drei Gruppen aufgeteilt: zwei Gruppen, die im Wechsel ein Wochenende frei bekommen, und eine, die jedes Wochenende frei bekommt. Für die zwei wechselnden Gruppen gilt, dass bei der 6-Tage-Woche das Pflegepersonal den Samstag und Sonntag und bei der 5-Tage-Woche von Freitag bis Montag (jeweils inkl.) frei bekommt. Die Gruppe, die jedes Wochenende frei hat, bekommt unabhängig von der Tage-Wochen-Regelung immer den Samstag und Sonntag frei. Die freien Tage sind harte Bedingungen im Modell, hierdurch ist dann auch der Sonntagsausgleichstag gesichert und muss im Modell nicht berücksichtigt werden.

Bemerkung 2.1:

Erste Simulationen haben ergeben, dass bei der 5-Tage-Woche die Bedingung, von Freitag bis Montag (jeweils inkl.) frei zu bekommen, zu hart ist. Daher wurde noch ein zusätzliches Modell für die 5-Tage-Woche mit einer alternativen Wochenendregelung erstellt. In dieser Alternative werden als fixe Wochenendtage nur der Samstag und Sonntag gegeben. Die übrigen zwei Tage werden zufällig auf die anderen 12 Tage verteilt. Diese Tage werden allerdings möglichst (unter Berücksichtigung anderer harter Bedingungen) so verteilt, dass keine Pflegedefizite entstehen.

2.9 Krankheitsfälle und Urlaubsregelung

Ein großes Problem in der Personaleinsatzplanung ist, dass einzelne Pflegekräfte unplanmäßig („von einem Tag auf den anderen“) erkranken und so nicht dienstbereit sein können. Der Krankheitsfall kann in der Regel nicht geplant werden, einige wenige andere Fälle einmal ausgeschlossen. Als weitere Erschwernis kommt hinzu, dass zu bestimmten Zeitpunkten mehr Personal erkrankt (Grippezeit) als sonst und hier ein gewisses Epidemieverhalten vorliegt.

Eine ähnliche Planungsproblematik hat man mit dem Urlaub von Pflegepersonal. Zwar weiß man im Voraus, wann das Pflegepersonal in Urlaub möchte¹, und laut Tarifvertrag gilt auch das gesamte Kalenderjahr als Urlaubszeitraum. Trotzdem kann man den Urlaub nicht einfach planen, wie er der Pflegeeinrichtung am liebsten ist, sondern muss weitestgehend auf die Urlaubswünsche des Pflegepersonals reagieren. Hierbei müssen soziale² als auch unternehmerische Gesichtspunkte beachtet werden; daher darf nicht beliebig viel Pflegepersonal auf einmal in Urlaub sein, sondern nur eine begrenzte Anzahl hiervon. Diese Anzahl wird über den Stellenanteil des Pflegepersonals bestimmt. Im Moment dürfen 4 Vollzeitkräfte auf einmal in Urlaub gehen; daher wäre es auch möglich, dass 8 Halbzzeitkräfte auf einmal in Urlaub gehen. Da sehr wenige Pflegekräfte auf einer Vollzeitstelle arbeiten, können sich bei 4 Vollzeitkräften schnell mal 7 bis 8 Personen gleichzeitig Urlaub nehmen. Des Weiteren hat jede Pflegekraft nur eine begrenzte Anzahl Urlaubstage im Jahr.

In der Regel wird in der klassischen Personalplanung zur Betrachtung des Arbeitsausfalls durch Krankheit und Urlaub angenommen, dass 20% der Arbeitszeit wegfällt. Dass diese Bilanzrechnung („Milchmädchenrechnung“) nicht sehr realitätsnah ist, kann am folgenden Beispiel gezeigt werden.

Beispiel:

Im Beispiel werden 5 Frühschichten an aufeinander folgenden Tagen betrachtet (Wochenende sei außer Acht gelassen), jede Schicht hat einen Pflegebedarf von 4 Pflegekräften. Zur Deckung des Pflegebedarfes hat man 5 Vollzeitkräfte. Nach der Bilanzrechnung hätte man genügend Pflegepersonal.

$$5 \text{ mal Frühschicht mit } 4 \text{ Pflegekräften} = 20 \text{ Schichtzeiten}$$

$$5 \text{ Vollzeit-Pflegekräfte} * 80\% = 20 \text{ Schichtzeiten}$$

Demnach würde in der Theorie kein Pflegedefizit entstehen. Das geht aber in der Praxis nur selten auf, dazu betrachte man die folgende Tabellen, in denen D für dienstbereit und K für Krank steht.

¹Es existieren bestimmte Zeiträume wie Sommerferien, Weihnachten, usw., die als Urlaubszeitraum von den meisten Pflegekräften präferiert werden.

²Hier muss im Besonderen auf Familien mit Kindern geachtet werden, welche Urlaub zu Zeiten benötigen, wo auch Schulferien sind. Des Weiteren sollte es nicht vorkommen, dass bestimmtes Personal immer den Wunschurlaub erhält und anderes nie.

Tabelle 2.7: Darstellung einer funktionierenden Bilanzrechnung

	Frühschicht 1	Frühschicht 2	Frühschicht 3	Frühschicht 4	Frühschicht 5
Pflegekraft 1	D	D	D	D	K
Pflegekraft 2	D	D	D	K	D
Pflegekraft 3	D	D	K	D	D
Pflegekraft 4	D	K	D	D	D
Pflegekraft 5	K	D	D	D	D

Tabelle 2.8: Darstellung einer nicht funktionierenden Bilanzrechnung

	Frühschicht 1	Frühschicht 2	Frühschicht 3	Frühschicht 4	Frühschicht 5
Pflegekraft 1	D	D	D	D	K
Pflegekraft 2	D	D	K	D	D
Pflegekraft 3	D	D	K	D	D
Pflegekraft 4	K	D	D	D	D
Pflegekraft 5	K	D	D	D	D

In beiden Darstellungen würde ein Arbeitsausfall von 20% bestehen, allerdings zeigt die zweite Darstellung, dass eine einfache Bilanzrechnung nicht ausreicht.

Um den Krankheitsfall und den Urlaub von Pflegekräften realitätsnäher zu modellieren, wird im Modell durch den Zufall bestimmt, ob eine Pflegekraft krank wird oder in den Urlaub will. Dazu wird jeweils die Wahrscheinlichkeit eines Zustandswechsels von „dienstbereit nach Krank“ und „dienstbereit nach Urlaub“ ermittelt. Dies wurde jeweils monatsweise gemacht, um so auch die Krankheits- und Urlaubsspitzen mit darzustellen. Die Wahrscheinlichkeiten sind in folgender Tabelle dargestellt.

Tabelle 2.9: Wahrscheinlichkeiten der Zustandsübergänge

	Aug.	Sep.	Okt.	Nov.	Dez.	Jan.	Feb.	Mär.	Apr.	Mai	Jun.	Jul.
Krank	1,17%	0,80%	1,96%	3,25%	2,76%	3,10%	3,44%	4,58%	0,84%	1,18%	3,15%	1,88%
Urlaub	2,12%	5,31%	3,10%	4,58%	6,71%	1,46%	1,35%	3,04%	3,00%	4,46%	3,72%	6,18%

Ist per Zufallsgenerator bestimmt, ob eine Pflegekraft krank wird bzw. in Urlaub will, wird über eine zweite Verteilung die Dauer des Arbeitsausfalls bestimmt (siehe Abb. 2.2 und Abb. 2.3).

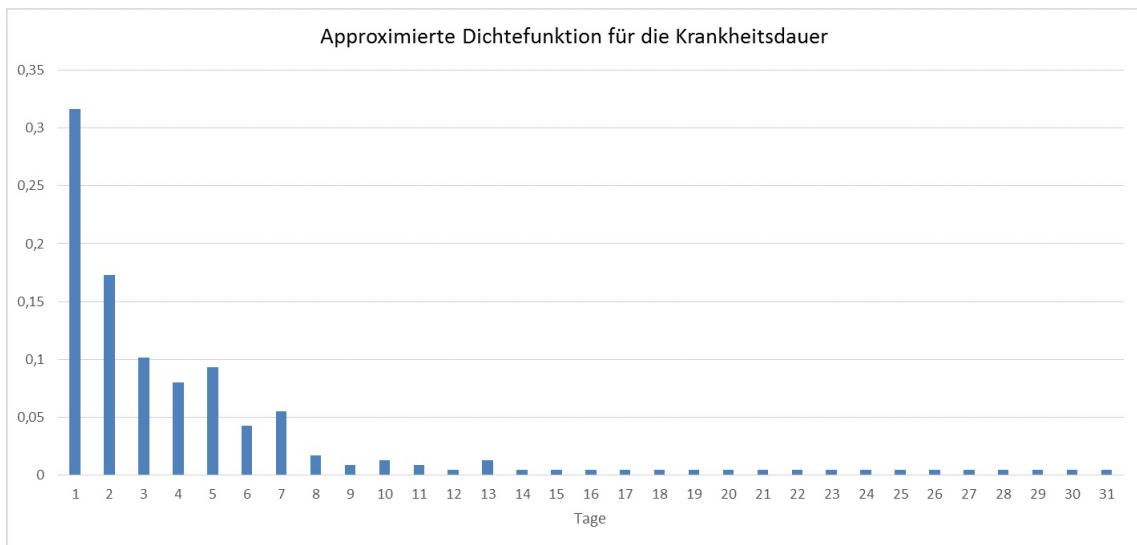


Abbildung 2.2: Dichtefunktion für die Modellierung der Krankheitsdauer

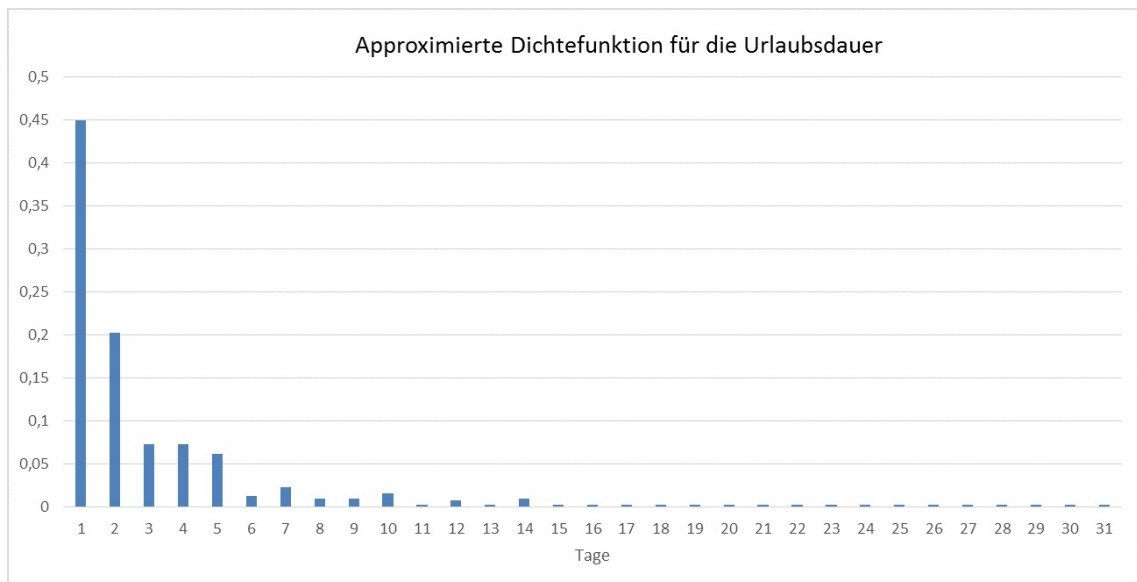


Abbildung 2.3: Dichtefunktionen für die Modellierung der Urlaubsdauer

Für die Urlaubsmodellierung gelten noch drei Besonderheiten. Erstens kann immer nur eine bestimmte Anzahl von Pflegepersonal, bestimmt über den Stellenanteil, Zeitgleich in den Urlaub; als Standardparameterwert wurde 4 Vollzeitkräfte eingestellt. Die zweite Besonderheit ist, wenn über ein freies Wochenende Urlaub genommen wird, dass diese freien Tage auf die Urlaubszeit dazu addiert werden. Drittens kann jede Pflegekraft nur eine begrenzte Anzahl von Urlaubstagen im Jahr nehmen.

Sowohl der Zustand „Krank“ als auch „Urlaub“ sind im Modell harte Bedingungen, sodass dieses Pflegepersonal nicht für die Absolvierung von Schichten eingesetzt werden kann.

3 Exemplarische Modelldarstellung mit Petri-Netzen

Im Folgenden soll die Modell-Spezifikation der Wochenendregelung bei einer 6-Tage-Woche exemplarisch mit Petri-Netzen nachvollzogen werden.

Da das Simulationsmodell am Donnerstag, den 1. August 2013 um 0:00 Uhr startet, müssen von diesem Zeitpunkt aus die Wochenend Intervalle betrachtet werden. Dies wird im folgenden Diagramm verdeutlicht.

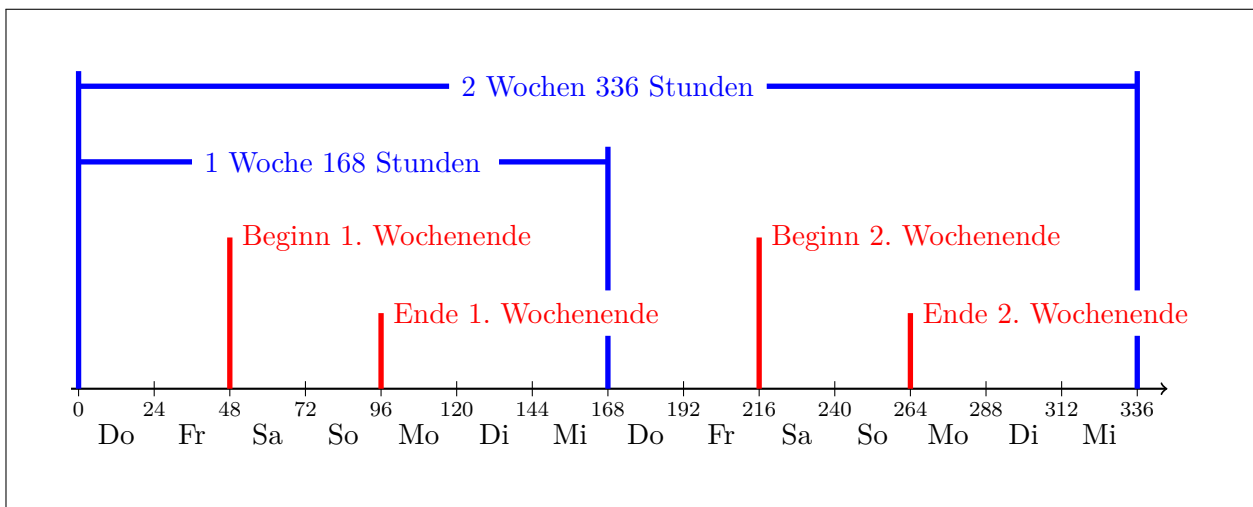


Abbildung 3.1: Wichtige Zeitpunkte für die Wochenendregelung

Aus diesen Daten kann man nun die drei Varianten von Wochenendregelungen (1. Wochenende frei, 2. Wochenende frei, jedes Wochenende frei) darstellen.

Tabelle 3.1: Zeitwerte für die Varianten von Wochenendregelung

Variante	Beginn Wochenende	Ende Wochenende	Intervall des Wochenendes
1. Wochenende frei	48	96	336
2. Wochenende frei	216	264	336
Jedes Wochenende frei	48	96	168

Stellt man nun die Variante „1. Wochenende frei“ als Petri-Netz dar, erhält man das Modell aus Abb. 3.2.

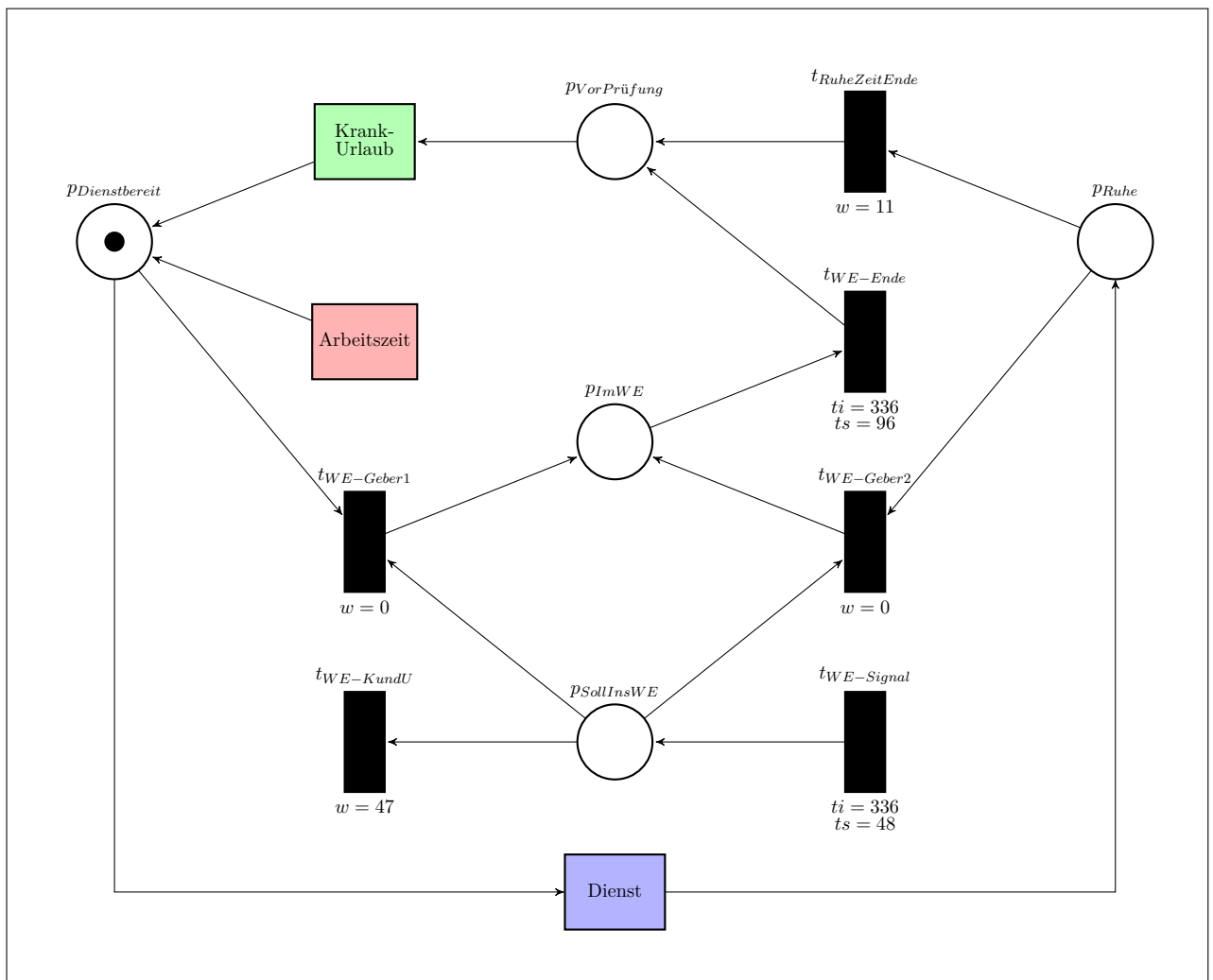


Abbildung 3.2: Petri-Netz-Modell bei „1. Wochenende frei“

In diesem Petri-Netz stellen die Blöcke „Dienst“, „Arbeitszeit“ und „Krank/Urlaub“ kompakte Vereinfachungen weiterer Komponenten des Gesamtmodells dar, die hier nicht weiter spezifiziert und erläutert werden sollen. Zudem wird unterstellt, dass sämtliche Pfeile mit 1 (Token) gewichtet sind, sodass der Einfachheit halber auf diese explizite Gewichtung verzichtet wird.

Die Bedeutung der Plätze und Transitionen des obigen Teilnetzes und ihr „Zusammenspiel“ soll im Folgenden ausführlich beschrieben werden, wobei zu beachten ist, dass hierbei jeweils *eine* konkrete Pflegekraft im Blickpunkt der Betrachtung steht:

Der Knoten $t_{\text{WE-Signal}}$ ist eine Transition mit Zeittakt¹, mit einem Taktstart von $ts = 48$ und einem Taktintervall von $ti = 336$; hierdurch wird abgebildet, dass der Beginn des ersten Wochenendes der Zeitpunkt 48 ist (von Freitag auf Samstag um 0:00 Uhr) und sich die Wochenenden im Rhythmus von 336 Stunden periodisch wiederholen. Diese Transition „feuert“ zu diesem Zeitpunkt

¹Näheres hierzu vgl. [KD15, S. 44 ff.]

Personalplanung im Pflegebereich

automatisch, d.h. übergibt einen Token (\bullet) an den Platz $p_{SollInsWE}$, der damit seinen Zustand ändert ($0 \rightarrow 1$ Token). Der Platz $p_{SollInsWE}$ zeigt dementsprechend an, ob die Pflegekraft ins Wochenende gehen soll (beim Zustand 1) oder nicht (beim Zustand 0).

Die Plätze $p_{Dienstbereit}$ und p_{Ruhe} zeigen an, ob die Pflegekraft aktuell dienstbereit bzw. sich in Ruhezeit befindet (Zustand 1) oder nicht (Zustand 0). Im Beispiel ist abzulesen, dass die Pflegekraft gegenwärtig dienstbereit ist. An dieser Stelle sei bereits angemerkt, dass der Modellablauf so konzipiert ist, dass stets genau einer der fünf Zustände „dienstbereit“, „im Dienst“, „in Ruhe“, „im Wochenende“ oder „Krank/Urlaub“ zutrifft.

Die beiden Transitionen $t_{WE-Geber1}$ und $t_{WE-Geber2}$ sind Transitionen mit Wartezeit¹. Da beide Transitionen die Wartezeit $w = 0$ haben, würden sie jeweils sofort feuern, sobald die Plätze $p_{SollInsWE}$ und $p_{Dienstbereit}$ bzw. p_{Ruhe} jeweils den Zustand 1 aufweisen. Hierdurch würde die Pflegekraft in den Zustand „im Wochenende“ versetzt; dies entspricht genau dem Zustand 1 von Platz p_{ImWE} . Durch die Transition $t_{WE-Ende}$ mit Zeittakt ($ts = 96, ti = 336$) wird die Pflegekraft zum jeweiligen Zeitpunkt (von Sonntag auf Montag um 0:00 Uhr) „aus dem Wochenende geholt“, indem der Zustand von Platz p_{ImWE} in den Zustand 0 geändert wird, falls er sich im Zustand 1 befindet. In diesem Fall wird der Zustand von Platz $p_{Vorprüfung}$ erhöht. Letzteres geschieht auch dann, wenn die Transition $t_{RuheZeitEnde}$ feuert. Dies ist genau dann der Fall, wenn der Platz p_{Ruhe} in den Zustand 1 versetzt wird, allerdings mit einer Wartezeit von 11 Stunden. Hierdurch wird abgesichert, dass eine Ruhezeit von 11 Stunden eingehalten wird (vgl. Abschn. 2.7).

Sobald der Platz $p_{Vorprüfung}$ in den Zustand 1 versetzt ist, findet im Block „Krank/Urlaub“ die Überprüfung statt, ob die Pflegekraft in den Zustand „Krank“ bzw. „Urlaub“ übergeht (vgl. Abschn. 2.9).

Der Knoten $t_{WE-KundU}$ ist eine Transition mit Wartezeit ($w = 47$), wodurch der „Sonderfall“ Berücksichtigung findet, dass die Pflegekraft übers Wochenende krank oder im Urlaub ist, sodass das permanente Wochenendsignal (weder $t_{WE-Geber1}$ noch $t_{WE-Geber2}$ feuern, sodass Platz $p_{SollInsWE}$ im „Sonderzustand“ 1 verharrt) beendet und der Platz $p_{SollInsWE}$ in den „Normalzustand“ 0 zurückgesetzt wird

In Abb. 3.3 und Abb. 3.4 sind die Varianten „2. Wochenende Frei“ und „Jedes Wochenende Frei“ dargestellt.

Im folgenden Abschnitt soll nun eine Beispiel-Anwendung des Modells gegeben werden.

¹Näheres hierzu vgl. [KD15, S. 47 ff.].

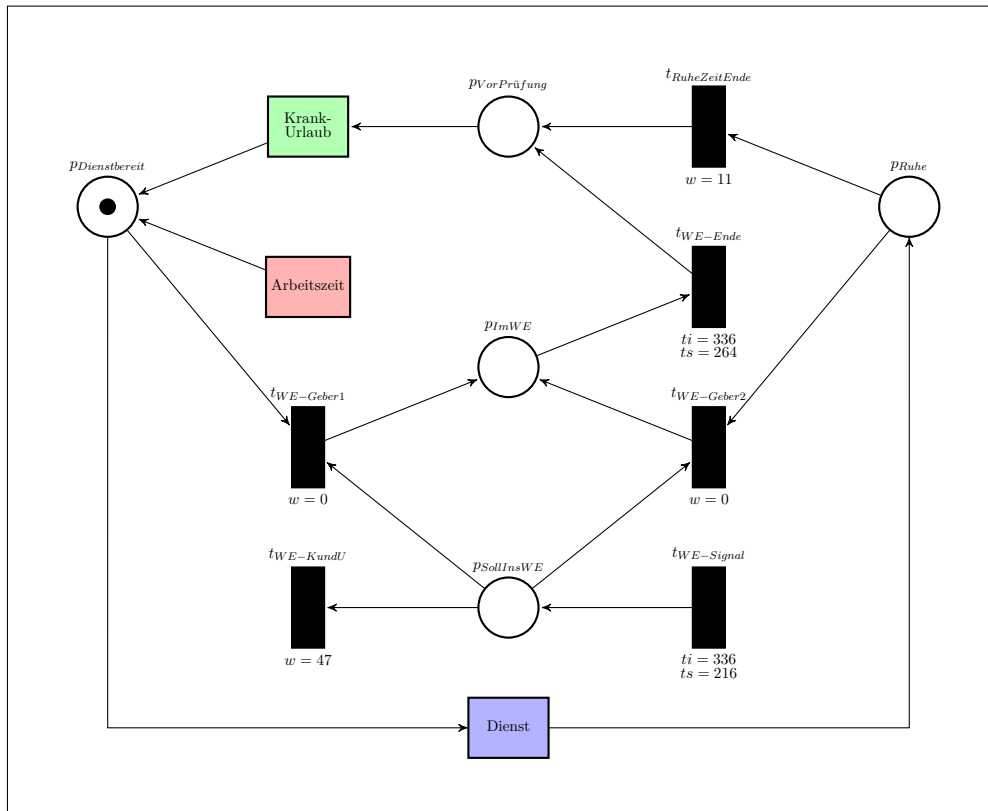


Abbildung 3.3: Petri-Netz-Modell bei „2. Wochenende frei“

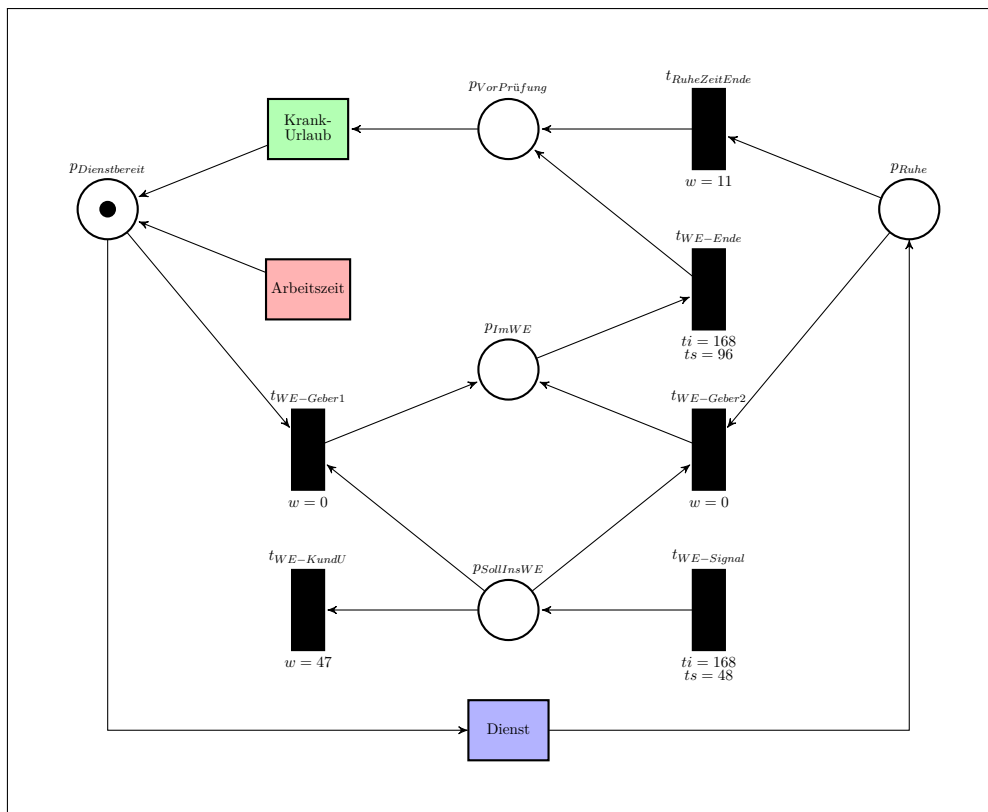


Abbildung 3.4: Petri-Netz-Modell bei „jedes Wochenende frei“

4 Beispiel-Anwendung

Da es sich bei der Simulation um eine stochastische Simulation handelt, müssen hinreichend viele Simulationen gemacht werden, um eine interpretierbare Aussage zu erhalten. Aus den Pflegedefiziten der Simulationen können Verteilungsfunktionen approximiert werden, welche dann miteinander verglichen werden können.

Im Folgenden wird an Abb. 4.1 erläutert, wie die Verteilungsfunktionen interpretiert werden sollten.

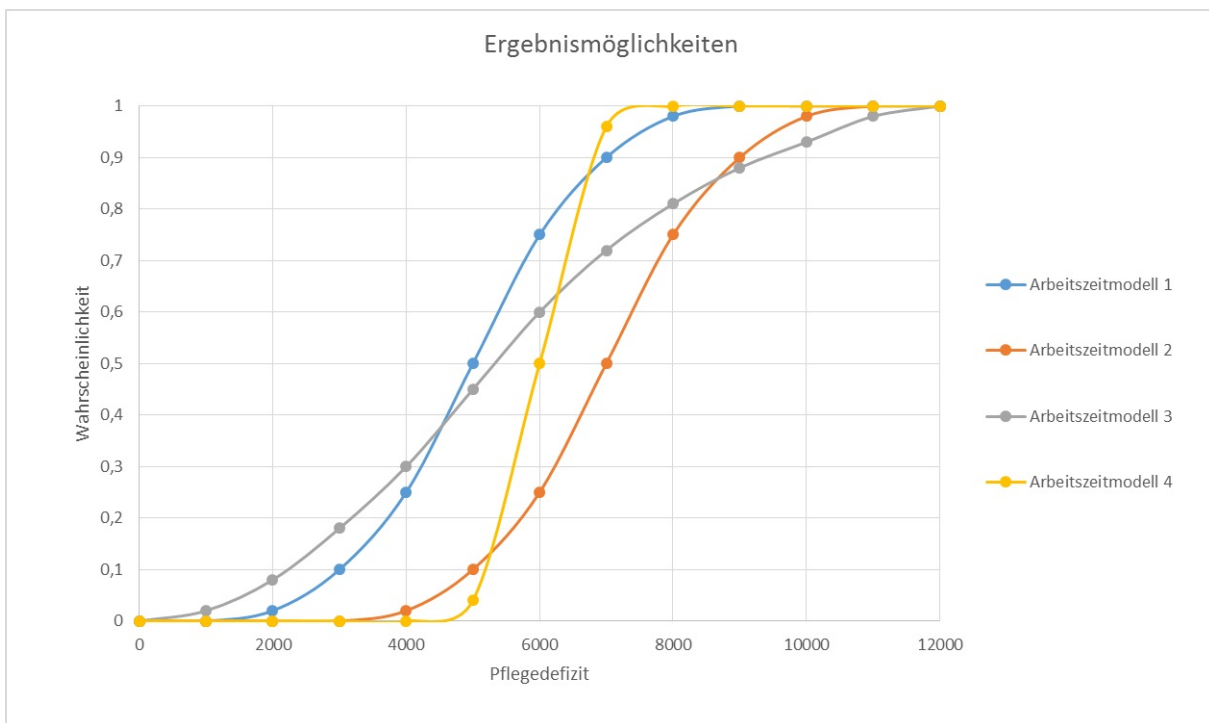


Abbildung 4.1: Fingiertes Beispiel für Simulationsergebnisse

Generell gilt: je weiter links eine Verteilung liegt, desto geringer wird das Pflegedefizit sein, und daher sind solche Arbeitszeitmodelle zu bevorzugen. Vergleicht man Arbeitszeitmodell 1 mit Arbeitszeitmodell 2, dann ist ersichtlich, dass Modell 1 in allen Punkten weiter links liegt als Modell 2; daher ist in einer solchen Situation Modell 1 immer Modell 2 vorzuziehen; man spricht von einer sogenannten „Wahrscheinlichkeitsdominanz“.

Sollten hingegen ein oder mehrere Schnittpunkte zwischen zwei Verteilungen bestehen, ist eine Empfehlung nicht mehr so eindeutig zu bestimmen. Vergleicht man Arbeitszeitmodell 1 und Arbeitszeitmodell 3 miteinander, dann ist die Wahrscheinlichkeit für „gute Jahre“ mit einem geringen Pflegedefizit bei Modell 3 größer, allerdings ist auch die Wahrscheinlichkeit für „schlechte Jahre“ größer als im Vergleich zu Modell 1.

Des Weiteren kann man Aussagen darüber machen, ob ein Arbeitszeitmodell stabil ist (das jährliche Pflegedefizit ist ungefähr immer gleich) oder ob es instabil ist (große Schwankungen im jährlichen Pflegedefizit); dies bezeichnet man auch als die „Streuung“ einer Verteilung. Das Arbeitszeitmodell 4 ist ein stabiles, da die Verteilungsfunktion einen sehr steilen Anstieg hat bzw. sich in einem kleinen Intervall realisiert. Dagegen ist das Arbeitszeitmodell 3 ein instabiles, da es einen sehr flachen Anstieg hat bzw. das Realisationsintervall sehr groß ausfällt.

Im Folgenden sollen die zwei Arbeitszeitkonzepte, 6-Tage-Woche mit drei Schichten und 5-Tage-Woche mit vier Schichten und flexiblem Wochenende (vgl. Bemerkung 2.1), miteinander verglichen werden. Dazu wird das folgende fiktive Pflegepersonal verwendet.

Tabelle 4.1: Parameterliste des fiktiven Pflegepersonals

Simulations ID	Berufsqualifikation	Stelle	Wochenendregelung	Anzahl Urlaubstage
pflegekraft0	Pflegefachkraft (m/w)	1,00	1. Wochenende	36
pflegekraft1	Pflegefachkraft (m/w)	1,00	1. Wochenende	36
pflegekraft2	Pflegefachkraft (m/w)	1,00	1. Wochenende	36
pflegekraft3	Pflegefachkraft (m/w)	1,00	1. Wochenende	36
pflegekraft4	Pflegefachkraft (m/w)	0,50	1. Wochenende	18
pflegekraft5	Pflegefachkraft (m/w)	0,50	1. Wochenende	18
pflegekraft6	Pflegefachkraft (m/w)	1,00	2. Wochenende	36
pflegekraft7	Pflegefachkraft (m/w)	1,00	2. Wochenende	36
pflegekraft8	Pflegefachkraft (m/w)	1,00	2. Wochenende	36
pflegekraft9	Pflegefachkraft (m/w)	1,00	2. Wochenende	36
pflegekraft10	Pflegefachkraft (m/w)	0,50	2. Wochenende	18
pflegekraft11	Pflegefachkraft (m/w)	0,50	2. Wochenende	18

Bemerkung 4.1:

Das Ziel der Simulation ist es, unterschiedliche Arbeitszeitmodelle miteinander zu vergleichen. Dabei muss beachtet werden, dass unterschiedliche Schichtzeiten in den Modellen gelten. Bei der 6-Tage-Woche sind die Schichten kürzer als bei der 5-Tage-Woche. Daher ist, bei „gleicher“ Sollbesetzung, die tägliche Arbeitszeit bei der 5-Tage-Woche größer. Dies würde im Vergleich die 5-Tage-Konzepte benachteiligen; daher müssen hier die Sollbesetzungen angepasst werden. Was für Sollbesetzungen in der Praxis anwendbar sind, aufgrund der Stationsorganisation, müsste durch „Expertenmeinung“ bestimmt werden. Hierfür sollten mehrere verschiedene Sollbesetzungen simuliert werden.

Es werden folgende fiktive Sollbesetzungen verwendet.

Tabelle 4.2: Fiktive Sollbesetzung bei 6-Tage-Woche und drei Schichten

Schicht	Pflegefachkraft	Arbeitsstunden
Früh	3	21
Spät	2	14
Nacht	1	9,25
Arbeitsstunden	44,25	

Tabelle 4.3: Fiktive Sollbesetzung bei 5-Tage-Woche und vier Schichten (1)

Schicht	Pflegefachkraft	Arbeitsstunden
Früh	2	15,4
Tag	2	15,4
Spät	1	7,7
Nacht	1	9,25
Arbeitsstunden	47,75	

Tabelle 4.4: Fiktive Sollbesetzung bei 5-Tage-Woche und vier Schichten (2)

Schicht	Pflegefachkraft	Arbeitsstunden
Früh	1	7,7
Tag	2	15,4
Spät	1	7,7
Nacht	1	9,25
Arbeitsstunden	40,05	

Des Weiteren soll maximal eine (1) Vollzeitkraft zeitgleich in den Urlaub gehen dürfen.

Im Folgenden werden einige Simulationsergebnisse betrachtet.

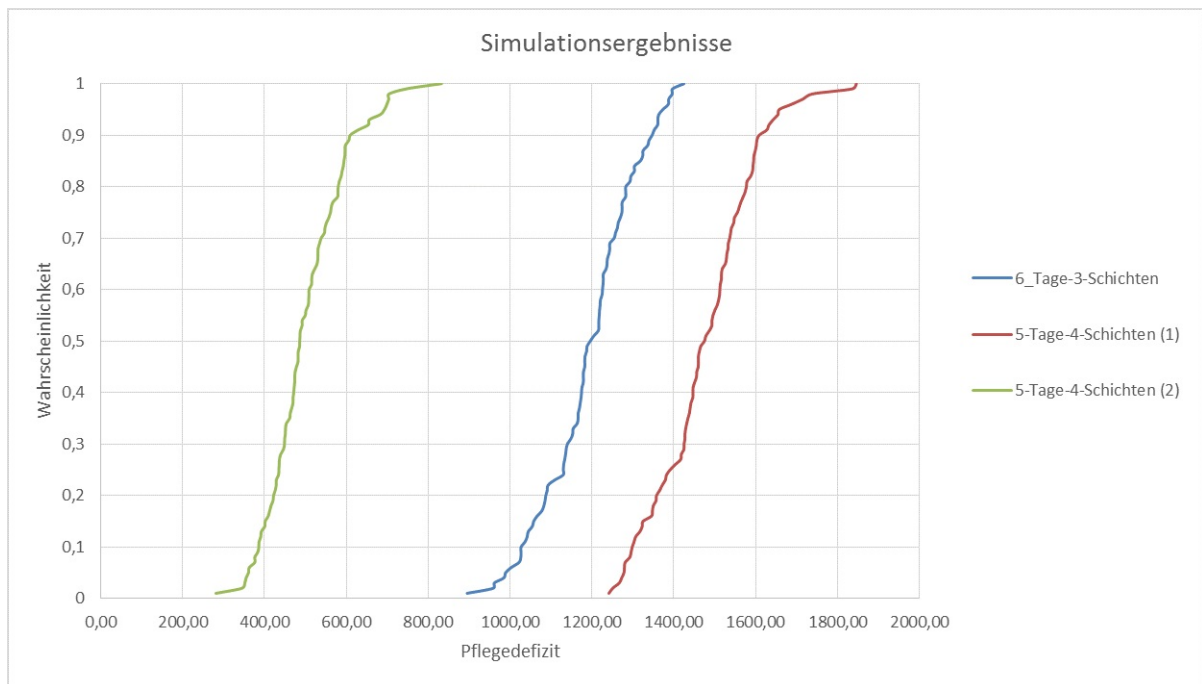


Abbildung 4.2: Simulationsergebnisse

Tabelle 4.5: Verteilungsparameter der Simulationsergebnisse

Parameter	6-Tage-Woche und drei Schichten	5-Tage-Woche und vier Schichten (1)	5-Tage-Woche und vier Schichten (2)
Minimum	895,00	1241,40	281,25
25% Quantil	1130,56	1400,38	435,71
Erwartungswert	1194,97	1478,33	502,28
75% Quantil	1273,38	1556,73	561,70
Maximum	1424,75	1846,10	832,45
Spannweite	529,75	604,70	551,20
Standardabweichung	112,76	123,23	96,64

6-Tage-Woche mit drei Schichten

Dies ist das Modell, welches eine „Istsituation der Referenzstation“ darstellen soll. Das erwartete Jahres-Pflegedefizit liegt bei 1194,97 Stunden mit einer Standardabweichung von 112,76 Stunden.

5-Tage-Woche mit vier Schichten (1)

Dieses Arbeitszeitmodell wird von der „Istsituation“ dominiert. Bei der Simulation ergab sich ein Erwartungswert von 1478,33 Stunden für das Pflegedefizit. Die Standardabweichung beträgt 123,23 Stunden, demnach ist das simulierte Arbeitszeitkonzept weniger stabil als die „Istsituation“. Demnach ist dieses Arbeitszeitmodell nicht empfehlenswert, wenn der Pflegekräftebedarf als alleiniger Maßstab zugrundegelegt wird.

Dieses erhöhte Pflegedefizit dürfte im Wesentlichen auf die erhöhte tägliche Arbeitszeit zurückzuführen sein, denn bei der „Istsituation“ beträgt sie 44,25 Stunden und bei diesem Modell 47,75 Stunden.

5-Tage-Woche mit vier Schichten (2)

Dieses Arbeitszeitmodell dominiert die „Istsituation“. Bei der Simulation ergab sich ein Erwartungswert von 502,28 Stunden für das Pflegedefizit. Die Standardabweichung beträgt 96,64 Stunden, demnach ist das simulierte Arbeitszeitkonzept auch noch stabiler als die „Istsituation“.

Dieses geringere Pflegedefizit ist auf die Verringerung der täglichen Arbeitszeit zurückzuführen, denn bei der „Istsituation“ beträgt sie 44,25 Stunden und bei diesem Modell 40,05 Stunden. Hier muss eine kritische Prüfung von Seiten eines „Experten“ im Hinblick auf die Verwendbarkeit der Sollbesetzungen durchgeführt werden!

Wäre die angenommene Sollbesetzung verwendbar, dann ist dieses Arbeitszeitmodell empfehlenswert, wenn der Pflegekräftebedarf als alleiniger Maßstab zugrundegelegt wird.

5 Ausblick

Die hier vorgestellten Untersuchungen sind als Vorarbeit für das Projekt FiliP [Fil15] ausgelegt gewesen. Der neue Aspekt im Projekt FiliP ist die „Alterung“ der Mitarbeiterschaft einer Pflegeeinrichtung. Um die Beschäftigungsfähigkeit der Mitarbeiterschaft möglichst lange zu erhalten, müssen Strategien entwickelt werden, um die Belastungsfaktoren zu reduzieren. Da einer der Hauptbelastungsgründe die Schichtarbeit und ihre Folgen sind, wäre es ein lohnendes Ziel, ein Arbeitszeitkonzept zu finden, das sich an den Alterungsprozess der Mitarbeiter/innen anpasst bzw. diesen signifikant berücksichtigt.

Hierfür muss unter anderem eine Pflegebedarfsabschätzung durchgeführt werden, um den Pflegebedarf nicht nur pro Tag, sondern im Verlauf des Tages zu betrachten. Hierbei müssen verschiedene Patiententypen (unter anderem auch Alterstypen) ermittelt werden, die den typischen Patienten-Mix einer Station abbilden. Aus dem abgeschätzten Pflegebedarf können dann genauere Sollbesetzungen bestimmt werden.

Um Pflegeeinrichtungen eine solche Planung zu ermöglichen, soll ein OpenSource-Tool entwickelt werden, mit dem einerseits der Pflegebedarf abgeschätzt werden kann und andererseits Arbeitszeitkonzepte a priori erprobt werden können. Dies soll jeweils durch Simulation erfolgen, die dann zur Entscheidungsunterstützung dienen soll.

Literaturverzeichnis

- [BC14] BURKE, E.K. und CURTOIS, T.: New approaches to nurse rostering benchmark instances. European Journal of Operations Research, 237:71–81, 2014.
- [BDBV04] BURKE, E.K., DE CAUSMAEKER, P., BERGHE, G.V. und VAN LANDEGHEM, H.: The state of the art of nurse rostering. Journal of Scheduling, 7:441–499, 2004.
- [BKK⁺14] BACHMANN, B., KLEINE-DÖPKE, T., KRUSE, H.-J., OCHEL, L. und PROSS, S. : Petri-Netz-Formalismen und Lösungsansätze für allgemeine Konfliktsituationen bei Feuerprozessen in Petri-Netz-Modellen. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Band 2, 2014.
- [Fil15] FILIP: Flexible und Intelligente Pflegepersonalplanung für ein demografie-festes Krankenhaus. Fachhochschule Bielefeld, 2015. <http://www.fh-bielefeld.de/forschung/forschungsprojekte/aktuelle-projekte-fb-3/nauerth-filip>.
- [Hac09] HACKMANN, T.: Arbeitsmarkt Pflege. Bestimmung der zukünftigen Altenpflegekräfte unter Berücksichtigung der Berufsverweildauer. FZG Diskussionsbeiträge, 2009.
- [Hie09] HIEN, W.: Pflege bis 67? Die gesundheitliche Situation älterer Pflegekräfte. Mabuse, 2009.
- [HM05] HASSELHORN, H.-M. und MÜLLER, BERND L.: Arbeitsbelastung und -beanspruchung bei Pflegepersonal in Europa - Ergebnisse der NEXT-Studie. In: BADURA, B., SCHELLSCHMIDT, H. und VETTER, C. (Herausgeber): Fehlzeiten-Report 2004, Seiten 21–50. Heidelberg: Springer, 2005.
- [KD15] KLEINE-DÖPKE, T.: Simulation und Optimierung der Personalplanung im Pflegebereich von Krankenhäuser durch Petri-Netz-Modelle in Modelica. Masterarbeit im Studiengang Optimierung und Simulation der Fachhochschule Bielefeld, 2015.
- [Pro13] PROSS, S.: Hybrid Modeling and Optimization of Biological Processes. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Band 1, 2013.
- [Röm15] RÖMER, M.: Von Piloten und Pflegekräften. OR News, November 2015:6–10, 2015.
- [SGW12] SCHAFF, M., GRIES, A. und WINTER, A.: Modellierung und Evaluation der Pflegeprozesse in der Notfallversorgung durch agentenbasierte Simulation. In: 6. Konferenz Netzwerk Ersteinschätzung 2012, Universität Leipzig, 2012.

Mathematische Modellierung des Laubharkproblems als spezielle Ausprägung von allgemeinen Entsorgungsprozessen

Hermann-Josef Kruse¹

¹Fachbereich Ingenieurwissenschaften und Mathematik, Fachhochschule Bielefeld

Kontakt:

hermann-josef.kruse@fh-bielefeld.de

Zusammenfassung. Die jährliche Laubentsorgung in Gärten und Parks stellt ein „populäres“ Beispiel für allgemeine Entsorgungsprozesse dar. An dieser „für jedermann“ nachvollziehbaren Problemstellung soll exemplarisch die systematische Vorgehensweise bei der Entwicklung eines mathematischen Modells für Praxisprobleme demonstriert werden, indem insbesondere auch auf die Komplexität dieses vermeintlich simplen Alltagsproblems eingegangen wird.

Schlüsselwörter. Laubharkproblem, Entsorgungsprozess

1 Herr K. und sein Laubharkproblem

Herr K. bewohnt mit seiner Frau ein Haus am Wald. Das Haus liegt ländlich und ruhig - sogar zu ruhig und einsam, wie viele seiner Freunde aus der nahe gelegenen Stadt meinen. Herr K. und seine Frau lieben aber die Ländlichkeit und ihr Haus. Viel mehr aber lieben sie den großen Garten mit den vielen Obst- und Zierbäumen und den Blumenbeeten. Hier findet Herr K. nach der täglichen Arbeit in der Stadt seine Erholung. Als leidenschaftlicher Hobbygärtner verbringt er sehr viel Zeit mit Gartenarbeiten, die er aber kaum als Arbeit, sondern vielmehr als Entspannung ansieht. Herr und Frau K. haben schon seit mehreren Jahren keine Urlaubsreise mehr unternommen, sondern verbringen die Ferien lieber zu Hause und in ihrem Garten. Man könnte also meinen, dass Herr und Frau K. für sich das Paradies auf Erden gefunden hätten – wäre da nicht der Herbst, und mit dem Herbst das viele Laub, welches Jahr für Jahr nicht nur von den eigenen Bäumen fällt, sondern auch aus dem Wald in den Garten hineingeweht wird und die großen Rasen- zu Laubflächen werden lässt. Und wenn Herr K. eines hasst, so ist es das jährliche Laubharken. Er selber würde das Laub allzu gerne ignorieren und es über Herbst und Winter im Garten liegen und das Laubproblem der Natur überlassen. Doch Frau K. besteht nun einmal darauf, dass der Garten zu jeder Jahreszeit einen ordentlichen Eindruck zu machen hat, und Laub auf dem Rasen verletzt diese „Gartenordnung“.



Und nun steht Herr K. wie in jedem Herbst wieder in seinem Garten und beschaut sich das stets wiederkehrende Laubproblem. Ausgerüstet ist er mit einer handelsüblichen Harke und einer Schubkarre als Laubwagen. Herr K. steht in der Mitte des Gartens und scheint gedankenversunken zu sein. In Wirklichkeit aber stellt er strategische Überlegungen an. Er begutachtet die Verteilung

Modellierung des Laubharkproblems

des Laubes über die gesamte Rasenfläche. Es gibt Bereiche, in denen sich sehr viel Laub angesammelt hat, und solche, in denen nur wenige Laubblätter anzutreffen sind. Herr K. steht vor der Wahl, wie er das Laub zu Haufen zusammenharken soll, in viele kleine oder in wenige große Haufen. Diese Entscheidung beeinflusst die sich daran anschließenden Transportprozesse mit dem Laubwagen zwischen den Laubhaufen und der Kompoststelle am Rande des Gartens. Bei kleinen Haufen bieten sich Rundtouren von Haufen zu Haufen und schließlich mit gefülltem Laubwagen zur Kompoststelle an. Bei großen Haufen könnten Pendeltouren von Kompoststelle zu den einzelnen Haufen und zurück günstiger sein. Darüber hinaus wäre der Einsatz einer größeren Harke zeitsparender, aber auch kräftezehrender als die Verwendung der kleinen, leichteren Harke. Nicht zuletzt muss bedacht werden, ob der Abtransport erst vorgenommen wird, wenn der Harkprozess komplett abgeschlossen ist, oder ob sich Harken und Abtransport abwechseln sollen.

Ob Herr K. sich alle angedeuteten Entscheidungsmöglichkeiten und die damit verbundenen Konsequenzen sorgfältig durchdacht haben wird, bevor er mit der Laubbeseitigung anfängt, wird dem Auge des Betrachters verborgen bleiben. Vermutlich wird Herr K. eine intuitive und auf langjährige Erfahrungen beruhende Laubharkstrategie anwenden. Auch bleibt dem Betrachter unklar, ob sich Herr K. ein möglichst zeit-, wege- oder kräftesparendes Vorgehen zum Ziel gesetzt hat. Allerdings dürfte die angerissene Laubharkproblematik eine(n) Mathematiker(in) ermuntern, weitere Details zu hinterfragen oder anzunehmen, ein mathematisches Modell daraus zu „stricken“, sich geeignete Lösungsverfahren auszudenken und diese in Bezug auf ihre Lösungsgüte hin zu überprüfen, um letztendlich Herrn K. eine fundierte Entscheidungsunterstützung anbieten zu können.

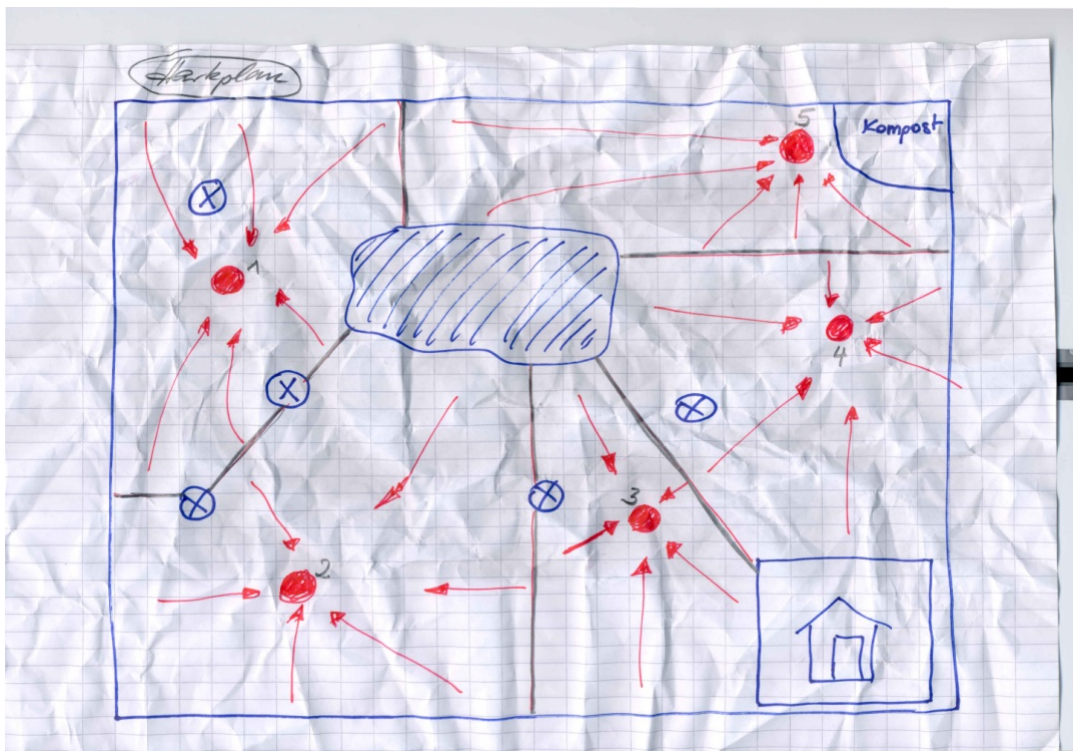


Abbildung 1.1: Wiederentdeckte Handskizze von Herrn K.

Das vorgestellte Laubharkproblem im Garten von Herrn K. lässt sich in den wesentlichen Grundzügen natürlich auch auf andere Gärten und sogar auf öffentliche Parkanlagen übertragen. Dabei handelt es sich offensichtlich um ein spezielles Entsorgungsproblem und gehört damit in den Bereich der Entsorgungslogistik¹. In der einschlägigen Literatur werden Entsorgungsprobleme in vielen verschiedenen Spezialbereichen behandelt, zumeist in Verbindung mit Abfall- und Müllentsorgung², und die zugehörigen Entsorgungsprozesse betreffen in erster Linie die Routenplanung³.

Bei dem hier behandelten Laubharkproblem als ein speziell ausgeprägtes Entsorgungsproblem steht zwar auch der Transportprozess zur Disposition, vielmehr aber ist das Augenmerk auf den Harkprozess gerichtet, und damit auf die Bestimmung der Stellen, an denen das Laub zusammengeharkt und zwischengelagert wird, bevor es von dort aus abtransportiert wird. Somit ergibt sich ein kombiniertes Problem der Standort- und Transportplanung. Bei genauerer Betrachtung lässt sich eine Spezialausprägung von sog. *Hub-Location-Problemen*⁴ erkennen.

In diesem Beitrag steht die Modellierung des Laubharkproblems im Vordergrund. Es werden verschiedene Abstraktionsstufen vorgestellt und Möglichkeiten der Berechnung von Hark- und Transportaufwänden entwickelt. Diese Modellierungsansätze liefern die Grundlage zur Entwicklung von effizienten Harkstrategien⁵, die in Verbindung mit Verfahren der Tourenplanung zur Lösung von Laubharkproblemen dienen.

2 Modellierungsansätze für Laubharkprobleme

Um das geschilderte Laubharkproblem von Herrn K. mit Hilfe von mathematischen Optimierungsverfahren einer Lösung zuführen zu können, bedarf es zunächst einer Abstraktion des Problems zwecks Bildung eines mathematischen Modells.

Die Modellbildung wird in mehreren Stufen erfolgen. Zunächst wird ein abstraktes Modell für den Garten geschaffen. Anschließend werden Überlegungen zur Festlegung einer geeigneten Zielgröße angestellt. Schließlich wird die Relevanz von problemspezifischen Nebenbedingungen und Annahmen diskutiert.

¹Zum Begriff „Entsorgungslogistik“ und der Abgrenzung zu anderen Teilgebieten der Logistik siehe u.a. [JBK13], [Stö93].

²Hierzu siehe u.a. [Sou96], [WZ98].

³Hierzu siehe u.a. [Bou02], [OvGG09], S. 44ff.

⁴Zum Begriff „Hub-Location-Problem“ und dessen spezielle Ausprägungen siehe u.a. [Cam96], [HP09], [Kli91], [Wag06].

⁵Konkrete Harkstrategien werden explizit in einem anderen Beitrag dieses Buches vorgestellt (siehe [KSD16]).

Modellierung des Laubharkproblems

In der folgenden Abbildung ist der Garten von Herrn K. skizziert. Man erkennt die Rasenfläche (lindgrün), den Teich (blau), das Gerätehaus (rot), den Kompost (dunkeloliv), einige Bäume (braun) und die gepflasterten Gehwege (grau). Natürlich sind Rasenflächen nicht in allen Gärten rechteckig, wie hier „zufällig“ im Garten von Herrn K., aber diese Gegebenheit wird die Modellierung des Gartens und die Veranschaulichung der noch zu entwickelnden Laubharkstrategien erleichtern.¹

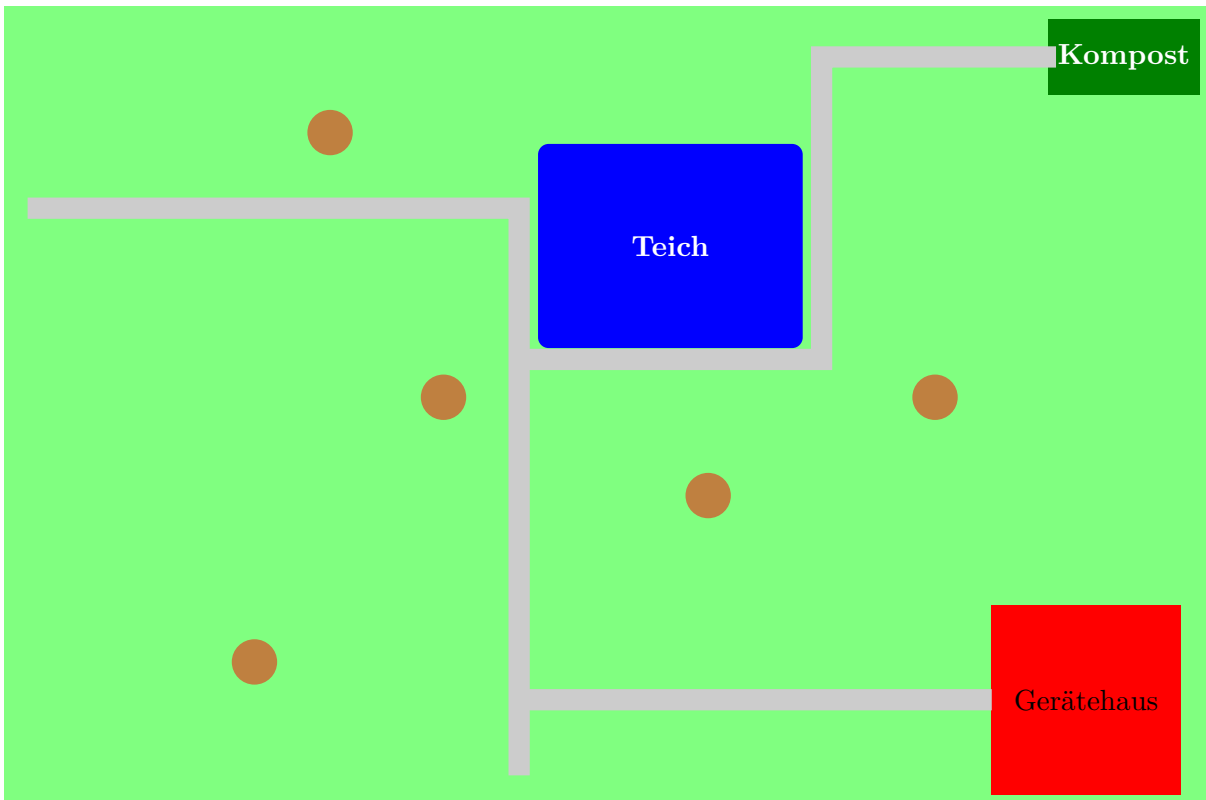


Abbildung 2.1: Skizze des Gartens von Herrn K.

2.1 Gartenmodell durch Matrixdarstellung

Als weitergehende Abstraktionsstufe² bietet sich die Rasterung der Rechteckfläche an (siehe Abb. 2.2). Hierbei entsteht eine Matrixform, bestehend aus m Zeilen und n Spalten. Die *Felder* in dieser Matrixdarstellung werden mit (i, j) bezeichnet, wobei i und j auch als *Koordinaten* des Feldes bezeichnet werden ($i = 1, \dots, m, j = 1, \dots, n$).

Wie fein- bzw. grobmaschig die Rasterdarstellung sein soll, bleibt dem Modellbauer überlassen.

¹Inwieweit diese Rechteckform die Eignung von Laubharkstrategien beeinflussen kann, wird später noch zu diskutieren sein. Entsprechende Überlegungen werden auch für Besonderheiten eines Gartens anzustellen sein (z.B. Existenz und Positionierung von Bäumen, Teichen, Gerätehaus, u.a.).

²Als erste Abstraktionsstufe muss bereits die Skizzierung des Gartens angesehen werden, zumal diese schon ein Modell der Realität darstellt. Streng genommen wäre sogar schon ein Foto des Gartens (z.B. aus der Vogelperspektive) ein Modell.

Je grober die Rasterung angelegt wird, um so mehr werden die realen Konturen des Gartens verloren gehen. Es wird also die „Kunst“ des Modellbauers sein, die Größe $m \times n$ der Matrix so einzurichten, dass die Laubharkstrategien, die anhand des Rastermodells getestet werden, den realen Ablauf eines Harkprozesses in seinen wesentlichen Grundzügen zweckdienlich widerspiegeln.

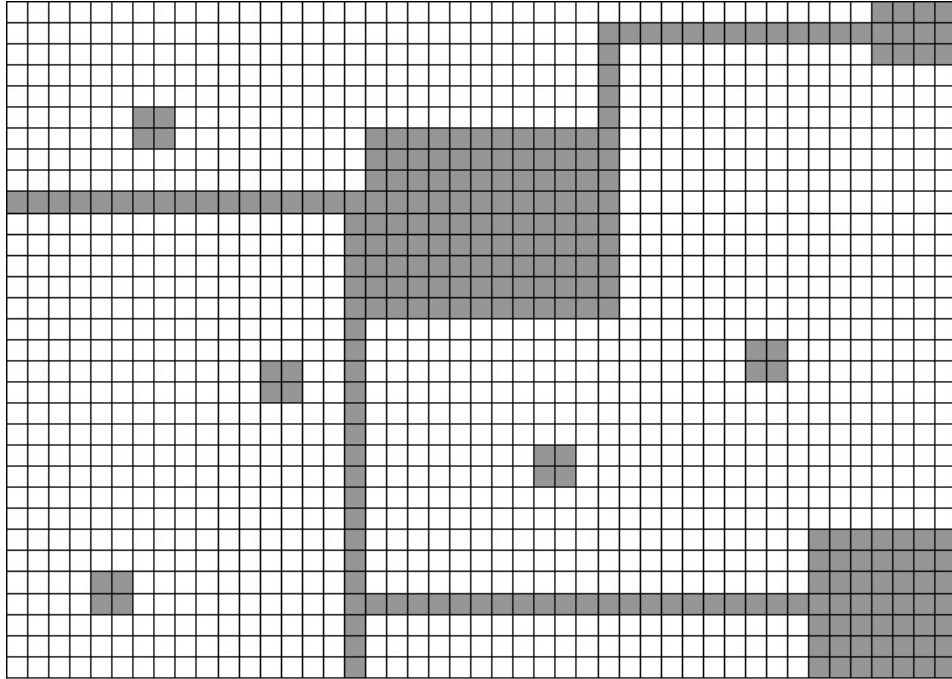


Abbildung 2.2: Gerasterte Gartendarstellung

Es wird sich als sehr praktisch erweisen, jedes Feld anstelle eines Koordinatenpaares (i, j) auch durch eine natürliche Zahl $a \in \mathbb{N}$ eindeutig zu beschreiben. Hierfür wird jedem Feld (i, j) die Zahl a gemäß

$$a = (i - 1) \cdot n + j \tag{2.1}$$

zugewiesen.¹

Nun ist es an der Zeit, sich dem Laub zu widmen, das sehr unterschiedlich im Garten und insbesondere – weil problemrelevant² – auf der Rasenfläche (und den Gehwegen) verteilt liegt. Genau genommen müsste man sich jetzt die Mühe machen, in jedem relevanten Feld (i, j) , das zur Rasenfläche (oder den Gehwegen) gehört, die Anzahl der Laubblätter zu zählen und diese Anzahl dem Feld (i, j) zuzuweisen. Eine derart genaue Laubbestandsaufnahme käme allerdings einer Einzelblattlese gleich, wonach sich das Laubharkproblem nicht mehr stellte. Daher werden die Laubmengen in den Feldern höchstens nach Augenschein geschätzt und registriert. Jedes Feld a bekommt

¹Man mache sich klar, dass es sich bei $(i, j) \rightarrow (i - 1) \cdot n + j$ um eine bijektive Abbildung handelt.

²Hierbei muss von vornherein festgelegt sein, welche Felder vom Laub befreit werden sollen. Dieses sind sicherlich die Felder der Rasenfläche, können womöglich aber auch die Felder der Gehwege sein, während die Felder, die zum Teich, zum Gerätehaus und zum Baumbestand zählen, keiner Laubentfernung durch Harken unterstellt werden sollen bzw. können.

Modellierung des Laubharkproblems

somit eine (wie auch immer ermittelte) *Laubmenge* $M(a)$ zugeordnet.¹ Diese Laubmenge wird in *Laubmengeneinheiten* [ME] gemessen.² Für jede Laubmenge gilt sinnvollerweise die Nichtnegativitätsbedingung:

$$M(a) \geq 0. \quad (2.2)$$

Dabei kann die Laubmenge je nach Gutdünken des Modellbauers bzw. nach Präzision der Laubmengenermittlung als reelle oder als ganze Zahl zugelassen werden. Der zweite Fall, d.h. $M(a) \in \mathbb{N}_0$, bietet sich an, wenn zur Ermittlung der Laubmengen feste Klassengrößen vorgegeben sind, etwa $M(a) = 0$ für den Fall, dass kein Laub im Feld a vorhanden ist, und $M(a) = 1, 2, 3$ usw. für den Sachverhalt, dass die Laubmenge $M(a)$ mit einem Zug bzw. mit zwei bzw. drei Zügen aus dem Feld a in ein benachbartes Feld geharkt werden kann. Der Einfachheit halber soll im Folgenden zunächst die ganzzahlige Variante benutzt, später aber auch die allgemeinere Variante $M(a) \in \mathbb{R}$ diskutiert werden.

In der Matrixdarstellung des Gartens wird nun in jedes Feld $a = (i, j)$ der Wert der zugehörigen Laubmenge $M(a) = M(i, j)$ geschrieben (siehe Abb. 2.3), wobei die Gleichung $M(a) = 0$ zwei Rückschlüsse zulässt: Zum einen kann es ein problemrelevantes Feld sein, welches lediglich kein Laub aufweist; zum anderen kann es sich um ein grundsätzlich auszuschließendes Feld handeln, etwa ein Teich- oder Gerätehausfeld. Die folgende Laubmengenmatrix M stellt einen mit Laub befallenen Garten dar, allerdings handelt es sich dabei um ein gegenüber dem Garten von Herrn K. vereinfachtes Beispiel ($m = 10, n = 20$). Dabei seien die grau unterlegten Nullfelder diejenigen, die im Harkprozess auszulassen sind.

1	2	0	3	3	2	4	3	3	2	1	2	3	4	4	3	2	1	0	0
2	2	3	2	3	2	3	3	4	3	2	3	3	5	4	4	3	1	0	0
2	4	5	3	2	2	2	2	3	1	1	1	2	4	2	5	4	3	1	0
3	6	0	4	3	2	2	2	1	0	0	0	0	2	3	6	0	4	2	3
2	5	4	4	3	5	4	3	1	0	0	0	0	1	3	6	5	3	2	2
1	2	2	2	1	5	8	6	2	0	0	0	0	2	4	3	3	3	2	2
1	4	4	3	2	5	0	7	4	2	3	4	4	5	0	4	0	0	0	0
1	4	0	5	3	6	5	6	4	3	2	1	2	1	3	3	0	0	0	0
1	2	3	5	2	3	3	4	2	1	1	2	2	2	2	1	0	0	0	0
1	2	1	2	1	1	1	1	0	0	0	1	2	3	2	2	0	0	0	0

Abbildung 2.3: Beispiel für einen Garten in Matrixdarstellung

Um die Nullelemente in der Matrix M bezüglich ihrer jeweiligen Bedeutung (laubleeres oder auszulassendes Feld) für spätere Berechnungen anhand des Zahlenwertes unterscheiden zu können (und

¹Die Laubmenge $M(a)$ für ein Feld $a = (i, j)$ wird bei Koordinatendarstellung mit $M(i, j)$ bezeichnet.

²Hierbei könnte es sich um die Anzahl der Laubblätter oder um 100 Gramm Laub oder um diejenige Laubmenge handeln, die bei einem Harkzug mit einer normierten Harke (durchschnittlich) bewegt werden kann. Hierauf wird später bei der Aufwandsberechnung für den Harkprozess noch näher eingegangen.

nicht unter Zuhilfenahme von grafischen Markierungen wie etwa Schattierungen), könnte vereinbart werden, die vom Laubharkprozess auszuschließenden Felder mit einem negativen Wert (z.B. -1) oder als Blank zu identifizieren.

In Anlehnung an das Harken in einem realen Garten wird der gesamte Harkprozess in Teilprozesse unterteilt, und zwar wird davon ausgegangen, dass die Laubmenge $M(a)$ eines Feldes a in ein benachbartes Feld b geharkt wird. Bei diesem Teilprozess werden die Laubmengen $M(a)$ und $M(b)$ verändert, und zwar gilt¹:

$$M_{neu}(b) = M_{alt}(b) + M_{alt}(a), M_{neu}(a) = 0. \quad (2.3)$$

Hierbei ist allerdings noch zu vereinbaren, was unter der „Nachbarschaft“ zwischen Feldern konkret verstanden werden soll. Hierzu bieten sich zwei Nachbarschaftsmodelle an.

Zwei Felder a und b heißen *benachbart*, wenn gilt:

$$a = (i, j) \wedge [b = (i, j - 1) \vee b = (i, j + 1) \vee b = (i - 1, j) \vee b = (i + 1, j)]. \quad (2.4)$$

Bei dieser Variante (a) lässt sich eine Laubmenge $M(a)$ immer nur senkrecht oder waagrecht, also in das nächste obere oder untere bzw. in das nächste linke oder rechte Feld harken (vgl. Abb. 2.4 links). Die nächste Variante (b) lässt zudem noch diagonales Harken und somit vier weitere Möglichkeiten zu (vgl. Abb. 2.4 rechts):

$$\begin{aligned} a = (i, j) \wedge [b = (i - 1, j - 1) \vee b = (i - 1, j) \vee b = (i - 1, j + 1) \vee b = (i, j - 1) \vee \\ b = (i, j + 1) \vee b = (i + 1, j - 1) \vee b = (i + 1, j) \vee b = (i + 1, j + 1)]. \end{aligned} \quad (2.5)$$

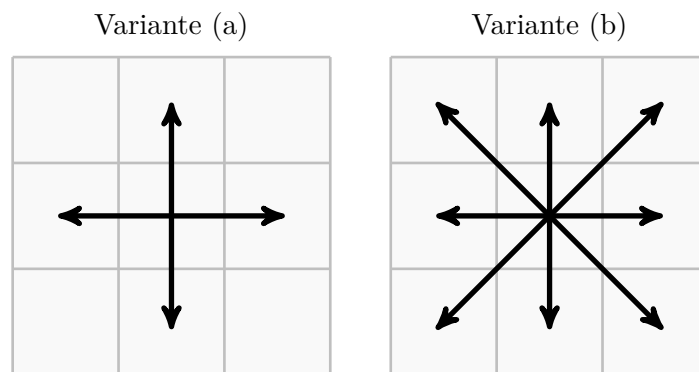


Abbildung 2.4: Nachbarschaftsvarianten

2.2 Gartenmodell durch Graphendarstellung

Ein in Matrixform abstrahierter Garten lässt sich anhand des Nachbarschaftsbegriffs auf kanonische Weise als *Graph* veranschaulichen. Jedes problemrelevante Feld a wird dabei als ein Knoten

¹In anderer Schreibweise: $M(b) \leftarrow M(b) + M(a), M(a) \leftarrow 0$.

Modellierung des Laubharkproblems

des Graphen aufgefasst; zwei benachbarte Felder a und b werden im Graph durch eine Kante $[a, b]$ verbunden. Der so entstehende Graph werde mit $G = (V, E)$ bezeichnet, wobei V die *Knotenmenge* und E die *Kantenmenge* des Graphen heißen. Da in der Praxis das Harken zwischen je zwei benachbarten Feldern a und b grundsätzlich in beide Richtungen möglich ist, handelt es sich bei G um einen ungerichteten Graph. Zudem bekommen die vom Harkprozess auszuschließenden Felder keine Knotenzuweisung. Jeder Knoten $a \in V$ von G wird mit der Laubmenge $M(a)$ des zugehörigen Feldes $a = (i, j)$ bewertet. Somit entsteht ein knotenbewerteter Graph $G = (V, E, M)$, wobei $M : E \rightarrow \mathbb{R}_{\geq}$ die *Knotenbewertung* von G heißt.

Das in Abb. 2.3 vorgestellte Gartenbeispiel in Matrixform erhält je nach Nachbarschaftsvariante die folgenden Darstellungen als Graph (siehe Abb. 2.5 bzw. Abb. 2.6), an denen unschwer zu erkennen ist, dass somit auch beliebige Gartenformen mittels Graphen abstrahiert werden können.

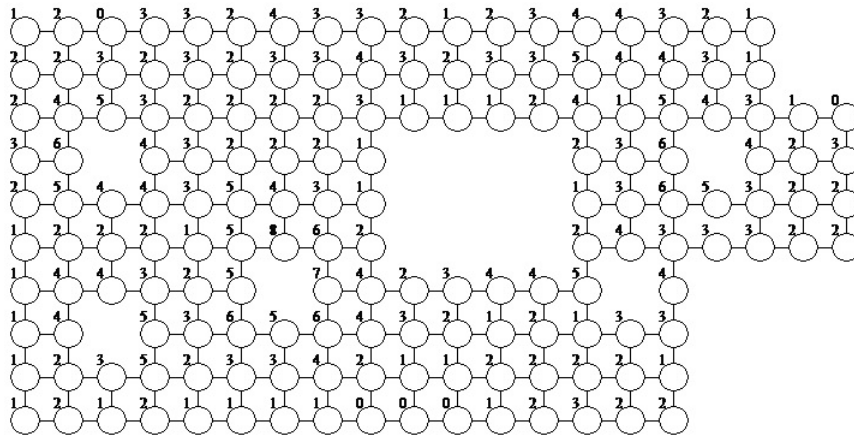


Abbildung 2.5: Graphendarstellung eines Gartens gemäß Nachbarschaftsvariante (a)

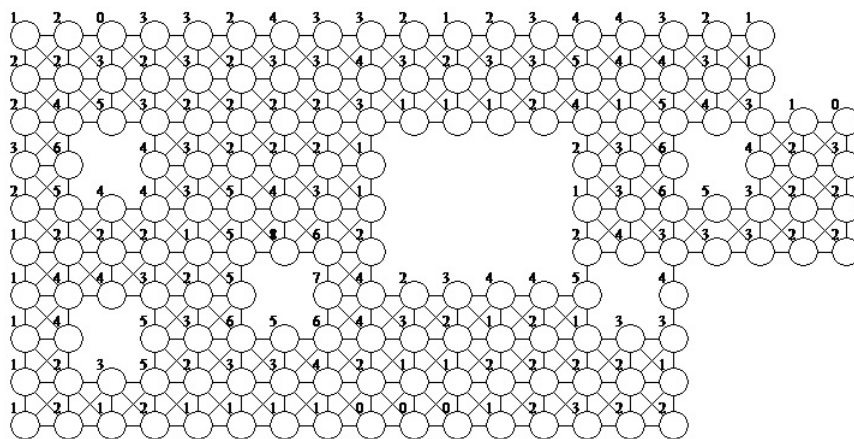


Abbildung 2.6: Graphendarstellung eines Gartens gemäß Nachbarschaftsvariante (b)

2.3 Aufwandsmodell für den Harkprozess

Nachdem nun Matrix- und Graphenmodelle zur abstrahierten Darstellung von Gärten eingeführt sind, stellt sich die ebenso wichtige Frage nach der Zielgröße, an der die Güte eines Entsorgungsprozesses gemessen werden soll. Hier ist wieder der Entscheidungsträger gefragt, denn wie einleitend bereits erwähnt legt Herr K. fest, ob er ein möglichst Zeit, Wege oder Kräfte sparendes Vorgehen anstrebt. Um hierfür viele Möglichkeiten offen zu lassen, soll im Folgenden sehr allgemein von „*Zeitkosten*“ gesprochen werden. Diese werden in der Dimension [ZE] für *Zeitkosteneinheit* gemessen. Hierbei kann es sich im konkreten Fall um Zeit [t] oder Kosten [GE] oder andere Messgrößen handeln. Es wird allerdings wichtig sein, darauf zu achten, dass alle Teilprozesse in derselben Dimension gemessen werden, sodass ein eindeutiges Gütemaß für den gesamten Entsorgungsprozess zur Verfügung steht.

Der gesamte Entsorgungsprozess besteht im Wesentlichen aus den drei Teilprozessen Harken, Aufladen und Abtransportieren des Laubes. Und so, wie es verschiedene Nachbarschaftsmodelle gibt, lassen sich auch verschiedene Ansätze zur Berechnung des jeweiligen Aufwandes dieser Teilprozesse finden.

Eine sehr einfache Berechnungsvorschrift ist es, den *Harkaufwand* $HA(a, b)$ eines Teilprozesses, d.h. den Aufwand für das Harken der Laubmenge $M(a)$ vom Feld a in das Feld b , proportional zur Laubmenge anzunehmen:

$$HA(a, b) = M(a) \cdot \alpha_H, \alpha_H > 0, \quad (2.6)$$

wobei α_H als *Harkaufwandsfaktor* oder kurz als *Harkparameter* bezeichnet werden soll, welcher angibt, wie viel Zeitkosten beim Harken einer Laubmengeneinheit [ME] von einem Feld a zu einem benachbarten Feld b entsteht. Dabei wird angenommen, dass der Harkparameter unabhängig von den Feldern eine konstante Größe darstellt. Der Parameter α_H hat in diesem Fall die Dimension [ZE/ME], sodass sich für den Harkaufwand die Dimension [ZE] ergibt.

Nach genauerer Betrachtung dieser simplen Harkaufwandsberechnung stellen sich allerdings erste Zweifel an deren Praxisnähe ein. Letztere lässt sich nämlich wohl nur unter idealisierten Voraussetzungen vertreten. Da wäre zunächst die Annahme der Feldunabhängigkeit zu untersuchen. Ist diese in der Praxis nicht gegeben, könnte allerdings der Übergang von der konstanten Größe α_H auf variable Größen $\alpha_H(a, b)$ als erste Verallgemeinerung der Berechnung des Harkaufwandes sinnvoll sein:

$$HA(a, b) = M(a) \cdot \alpha_H(a, b). \quad (2.7)$$

Somit ergibt sich (2.6) auf kanonische Weise als Spezialfall von (2.7), wenn $\alpha_H(a, b) \equiv \alpha_H$ für alle Felderpaare (a, b) gilt ($a \neq b$). Zudem wird $\alpha_H(a, a) = 0$ für alle Felder a vereinbart.

Weiterhin ist die Annahme der Proportionalität zweifelhaft. Hierbei wird unterstellt, dass sich der Aufwand für das Harken eines Blattes bei zwei Blättern verdoppelt, wenn exemplarisch für die Laubmengeneinheit [ME] ein Laubblatt angenommen wird. Praxisnäher aber dürfte die Unterstellung sein, dass ein Zug mit einer Harke nur unwesentlich von der damit bewegten Laubmenge abhängt, insbesondere im Hinblick auf die dafür verbrauchte Zeit. Erst wenn die Laubmenge eine kritische Höhe überschreitet, sodass mehr als ein einziger Zug zum Harken nötig ist, um die Laubmenge $M(a)$ von Feld a nach Feld b zu bewegen, ergibt sich ein signifikanter zeitlicher Mehraufwand. Um diesem Umstand im Modell gerecht zu werden, bietet es sich an, für diese kritische Größe eine positive reelle Zahl M_k einzuführen. Ist die Laubmenge $M(a)$ kleiner oder gleich diesem kritischen Wert, so wird genau ein Harkzug benötigt. Andernfalls wird man mit einem Zug nicht auskommen, sondern wird zwei oder mehrere Harkzüge benötigen. Diese reelle Zahl M_k soll *kritische Laubharkmenge* genannt werden.

Die Anzahl der benötigten Harkzüge bei gegebener Laubmenge $M(a)$ ergibt sich dann als nächste größere ganze Zahl zum Quotienten $M(a)/M_k$, sodass sich der Harkaufwand gegenüber (2.7) folgendermaßen ändert¹:

$$HA(a, b) = \lceil M(a)/M_k \rceil \cdot \alpha_H(a, b). \quad (2.8)$$

Man beachte nun aber, dass sich die Dimension der Harkparameter α_H bzw. $\alpha_H(a, b)$ geändert hat, zumal der Quotient $M(a)/M_k$ eine dimensionslose Größe darstellt ($[ME]/[ME]$). Danach müsste der Parameter α_H bzw. $\alpha_H(a, b)$ dieselbe Dimension wie der Harkaufwand $HA(a, b)$ besitzen, also in Zeitkosteneinheiten [ZE] gemessen sein. Diese Vereinbarung soll im Folgenden beibehalten werden, d.h. der Harkparameter α_H bzw. $\alpha_H(a, b)$ gibt an, wie viele Zeitkosteneinheiten für einen Harkzug benötigt werden, und zwar unabhängig von der Laubmenge, solange diese im Intervall zwischen 0 und der kritischen Laubmenge liegt, d.h. $0 < M(a) \leq M_k$.

Nachdem nun mehrere Varianten zur Modellierung des Harkaufwandes im Nachbarschaftsfall, d.h. für das Harken von einem Feld a auf ein Nachbarfeld b , vorgestellt worden sind, sollen weitergehende Überlegungen angestellt werden, wie sich diese Einzelaufwände für das Harken über mehrere Felder ($a \rightarrow b \rightarrow c \rightarrow d \rightarrow \dots$) übertragen (lassen). Hierzu wird exemplarisch ein kleiner Ausschnitt (die nordwestliche Ecke) aus dem obigen Beispiel-Garten entnommen.

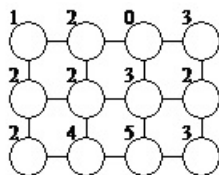


Abbildung 2.7: Nordwestlicher Ausschnitt aus Abb. 2.5

¹Hierbei sind $\lceil \cdot \rceil$ die Gaußklammern für die Aufrundung zur nächsten größeren ganzen Zahl.

Man nehme nun zunächst den idealisierten Fall an, dass die Laubmengen $M(a)$, die als Knotenbewertungen ersichtlich sind, mit der Anzahl der benötigten Harkzüge übereinstimmen. Es wird dabei also angenommen, dass jede Laubmenge $M(a)$ ein ganzzahliges Vielfaches der kritischen Laubmenge M_k bei konstantem Harkaufwandsparameter $\alpha_H = 1$ ist. Beispielsweise bedeutet $M(1, 2) = 2$, dass die Laubmenge aus dem Feld $a = (1, 2)$ mit genau zwei Harkzügen auf ein benachbartes Feld b zu harken ist. Somit ergibt sich $HA(a, b) = M(a) = 2$.

Es sei nun zunächst die Aufgabe gestellt, das gesamte Laub in diesem verkleinerten Garten auf *ein* ausgewähltes Feld zusammenzuharken, wobei sich die Frage nach dem gesamten Harkaufwand dafür stellt.

In diesem Zusammenhang stellt sich zunächst die Frage nach einer maximalen Laubmenge, die sich auf einem Feld „antürmen“ lässt und die es alleine schon aus natürlichen „Stabilitätsgründen“ für einen Laubhaufen gibt. Eine solche *maximale Laubmenge* könnte für jedes Feld anders ausfallen, d.h. $M(a) \leq \overline{M}(a)$, oder als gemeinsame Obergrenze für alle Felder festgelegt werden, d.h. $M(a) \leq \overline{M}$ für alle Felder a . Im Folgenden soll nur noch der einfache Fall einer gemeinsamen maximalen Laubmenge \overline{M} zur Anwendung kommen.

Um die obige Aufgabe erfüllen zu können (also *ein* Sammelhaufen), muss die maximale Laubmenge hinreichend groß sein; im obigen Beispiel muss offensichtlich gelten: $\overline{M} \geq 29$. Andernfalls käme man mit einem einzigen Sammelhaufen nicht aus.

Exemplarisch soll zunächst eine „Harkstrategie“ angewendet werden, wie sie in Abb. 2.8 als „Zickzack-Harken“ skizziert ist.

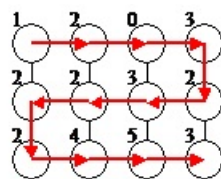


Abbildung 2.8: Zickzack-Harken (Variante 1)

Der gesamte Harkaufwand, der sich aus den in Tab. 2.1 aufgelisteten Einzelaufwänden (Anzahl der Harkzüge) zusammensetzt, beträgt dabei 124 Harkzüge.

Tabelle 2.1: Harkaufwand für Zickzack-Variante 1

Harken (von ... nach)	Aufwand (Anzahl Harkzüge)
(1,1) → (1,2)	1
(1,2) → (1,3)	3
(1,3) → (1,4)	3
(1,4) → (2,4)	6
(2,4) → (2,3)	8
(2,3) → (2,2)	11
(2,2) → (2,1)	13
(2,1) → (3,1)	15
(3,1) → (3,2)	17
(3,2) → (3,3)	21
(3,3) → (3,4)	26
Summe	124

Eine alternative „Zickzack-Strategie“ ist in Abb. 2.9 skizziert. Der gesamte Harkaufwand für diese Alternative beträgt 150 Harkzüge (vgl. Tab. 2.2).

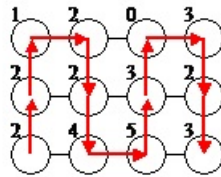


Abbildung 2.9: Zickzack-Harken (Variante 2)

Tabelle 2.2: Harkaufwand für Zickzack-Variante 2

Harken (von ... nach)	Aufwand (Anzahl Harkzüge)
(3,1) → (2,1)	2
(2,1) → (1,1)	4
(1,1) → (1,2)	5
(1,2) → (2,2)	7
(2,2) → (3,2)	9
(3,2) → (3,3)	13
(3,3) → (2,3)	18
(2,3) → (1,3)	21
(1,3) → (1,4)	21
(1,4) → (2,4)	24
(2,4) → (3,4)	26
Summe	150

Den beiden „Zickzack-Strategien“ wird in Abb. 2.10 eine weitere Harkmöglichkeit gegenübergestellt. Der gesamte Harkaufwand für diese 3. Variante beträgt 47 Harkzüge (vgl. Tab. 2.3).

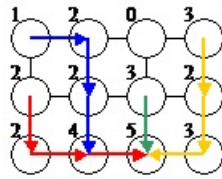


Abbildung 2.10: Hark-Variante 3

Tabelle 2.3: Harkaufwand für Variante 3

Harken (von ... nach)	Aufwand (Anzahl Harkzüge)
(1,1) → (1,2)	1
(1,2) → (2,2)	3
(2,2) → (3,2)	5
(2,1) → (3,1)	2
(3,1) → (3,2)	4
(3,2) → (3,3)	13
(2,3) → (3,3)	3
(1,4) → (2,4)	3
(2,4) → (3,4)	5
(3,4) → (3,3)	8
Summe	47

Anhand dieser Beispiele zeigt sich bereits sehr deutlich, dass der gesamte Harkaufwand entscheidend von der verwendeten „Harkstrategie“ abhängt. Dabei beeinflusst nicht nur die Festlegung der Harkreihenfolge ($a \rightarrow b \rightarrow c \rightarrow d \rightarrow \dots$) den gesamten Harkaufwand, sondern auch die Bestimmung desjenigen Feldes, auf dem der durch das Zusammenharken entstehende *Laubhaufen* errichtet werden soll.¹ Die sich daraus ergebende Frage nach einer effizienten (zeitkostengünstigen) Harkstrategie ist der zentrale Untersuchungsgegenstand eines weiteren Beitrages in diesem Buch (siehe [KSD16]).

Im Folgenden soll die Berechnung der Harkaufwände exemplarisch noch an einem weiteren Beispiel vollzogen werden, wobei vom idealisierten Fall, dass die Laubmengen $M(a)$ mit der Anzahl der benötigten Harkzüge genau übereinstimmen, Abstand genommen wird. Vielmehr wird nun angenommen, dass jede Laubmenge $M(a)$ als eine nicht-negative reelle Zahl vorkommt. Der Einfachheit halber soll aber weiterhin ein konstanter Harkparameter angenommen werden, und zwar o.B.d.A. $\alpha_H = 1$, sowie die kritische Menge $M_k = 1$ gesetzt sein. Beispielsweise bedeutet dann $M(1,2) = 2,5$, dass die Laubmenge aus dem Feld $a = (1,2)$ mit $[2,5] = 3$ Harkzügen auf ein benachbartes Feld b zu harken ist. Somit ergibt sich $HA(a,b) = \lceil M(a) \rceil = 3$.

Im folgenden Beispiel wird wieder nach der Zickzack-Variante 1 geharkt (vgl. Abb. 2.8), allerdings ergibt sich auf Grund der geänderten Zahlenwerte für die Laubmengen $M(a)$ ein anderes Ergebnis:

¹In größeren Gärten bestimmen auch noch weitere Kenngrößen die Suche nach effizienten Laubharkstrategien, z.B. die Anzahl und die Größe der zu errichtenden Laubhaufen.

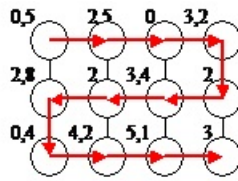


Abbildung 2.11: Zickzack-Harken (Variante 1) bei nicht-ganzzahligen Laubmengenwerten

Der gesamte Harkaufwand beträgt dabei 131 Harkzüge (vgl. Tab. 2.4). Für die Varianten 2 und 3 ergeben sich entsprechend 150 bzw. 48 Harkzüge als Gesamtaufwand.

Tabelle 2.4: Harkaufwand für Variante 1 bei nicht-ganzzahligen Laubmengenwerten

Harken (von ... nach)	Laubmenge $M(i, j)$	Aufwand (Anzahl Harkzüge)
(1,1) → (1,2)	0,5	1
(1,2) → (1,3)	3,0	3
(1,3) → (1,4)	3,0	3
(1,4) → (2,4)	6,2	7
(2,4) → (2,3)	8,2	9
(2,3) → (2,2)	11,6	12
(2,2) → (2,1)	13,6	14
(2,1) → (3,1)	16,4	17
(3,1) → (3,2)	16,8	17
(3,2) → (3,3)	21,0	21
(3,3) → (3,4)	26,1	27
Summe		131

2.4 Aufwandsmodell für den Transportprozess

Für das Aufwandsmodell des gesamten Laubharkproblems wird nun noch der Transportprozess einbezogen. Hierbei spielen zurückzulegende Transportwege eine entscheidende Rolle. Daher sollen noch Überlegungen zur Wegstreckemessung, sowohl bzgl. der Harkvorgänge als auch bzgl. des Transportprozesses, gemacht werden.

Bislang wird unterstellt, dass die Wegstrecke, welche die Laubmenge $M(a)$ beim Harken aus einem Feld a in ein benachbartes Feld b zurücklegt, der Einheitslänge für einen Harkzug entspricht. Diese Annahme lässt sich aber nur bei entsprechender Rasterung des Gartens vertreten. Je grober das Raster ausfällt, umso größer werden die Felder und umso größer wird die Wegstrecke, an der entlang eine Laubmenge aus Feld a in das benachbarte Feld b geharkt werden muss. Die hierbei zurückzulegenden Harkstrecken werden in Längeneinheiten [LE] gemessen. Hierbei kann 1 LE genau der Reichweite eines Harkzuges entsprechen oder aber in Metern oder Zentimetern angegeben werden, wobei der Abstand zwischen den beiden Mittelpunkten der quadratischen Felder gemessen wird, nicht nur der Einfachheit halber, sondern durchaus einer kanonischen Annahme folgend. Als

Abstandsmetriken zwischen zwei Feldmittelpunkten bieten sich die sog. *Manhattan-Metrik* (a) und die *Euklidische Metrik* (b) an (hierzu siehe Abb. 2.12).

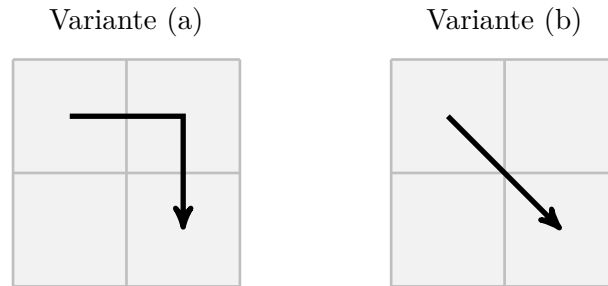


Abbildung 2.12: Abstandsmetriken zwischen Feldern

Im Folgenden wird zunächst die erste Variante, also die Manhattan-Metrik verwendet. Entsprechend sind die Kanten in der jeweiligen Graphendarstellung zu bewerten (hierzu siehe Abb. 2.13).

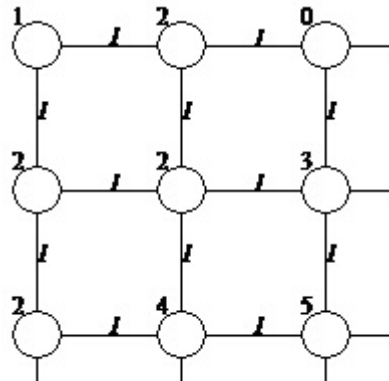


Abbildung 2.13: Ausschnitt einer Darstellung eines Gartens als knoten- und kantenbewerteter Graph gemäß Variante (a)

Natürlich können auch andere Kantenbewertungen als ein konstanter Wert $\beta \equiv 1$ [LE] verwendet werden, etwa $\beta(a, b)$ für benachbarte Felder bzw. Knoten a und b . In jedem Falle beträgt die Entfernung $d(a, b)$ zwischen zwei (nicht notwendig benachbarten) Feldern bzw. Knoten a und b einem kürzesten Weg im kantenbewerteten Graph G , d.h.

$$d(a, b) = \min_{w \in W(a, b)} L(w), \tag{2.9}$$

wobei $W(a, b)$ die Menge aller Wege zwischen a und b und $L(w)$ die Länge des Weges $w = \langle a, \kappa_1, \kappa_2, \dots, \kappa_r, b \rangle$ zwischen a und b mit Zwischenknoten κ_s ist ($s = 1, \dots, r$), d.h. $L(w) = \sum_{s=0, \dots, r} \beta(a_s, a_{s+1})$ mit $a_0 = a$ und $a_{r+1} = b$. Im Falle einer konstanten Kantenbewertung β beträgt die Entfernung zwischen zwei Feldern $a = (i, j)$ und $b = (k, l)$ demnach $\beta \cdot (|i - k| + |j - l|)$.

Die mögliche Einbeziehung der Entfernungen zwischen den Feldern in die Aufwandsberechnung für das Harken soll aber hier nur angedeutet bleiben. Viel wichtiger als für den Harkprozess sind

Modellierung des Laubharkproblems

die Entfernungen zwischen den Feldern nämlich auch für den Transportprozess. Hier allerdings dürfte die euklidische Metrik das passende Distanzmaß zur Entfernungsbestimmung sein. Dafür wäre allerdings vonnöten, dass die Lage der Feldmittelpunkte in einem kartesischen Koordinatensystem vorliegt. Wie dem auch sei, es wird im Folgenden angenommen, dass es eine Entfernungsmatrix D gibt, in der die Entfernungen zwischen je zwei Feldern (Knoten) vorliegen. Diese Entfernungsmatrix soll auch die Entfernungen der Felder zum Kompost (K) beinhalten. Alle Entfernungen haben die Dimension LE (Längeneinheit).

Der Transportaufwand ließe sich dann proportional zu den zurückgelegten Weglängen bestimmen. Bei einer Pendeltour zwischen dem Kompost K und einem Feld a ergibt sich im einfachsten Fall der Zeitaufwand

$$TA(a) = 2 \cdot D(K, a) \cdot \alpha_T, \quad (2.10)$$

wobei α_T als *Transportaufwandsfaktor* oder kurz als *Transportparameter* bezeichnet werden soll. Der Parameter α_T hat in diesem Fall die Dimension [ZE/LE], sodass sich für den Transportaufwand TA die Dimension [ZE] ergibt.

Allerdings können auch noch zusätzliche Aufwände eingerechnet werden, etwa die von der Laubmenge abhängige Auflade- und Abladezeit oder ein fixer Zeitaufwand (etwa für das Vorbereiten zum Aufladen oder zum Abladen), welcher unabhängig von der Entfernung und der Laubmenge ist:

$$TA(a) = 2 \cdot D(K, a) \cdot \alpha_T + \gamma \cdot M(a) + \sigma. \quad (2.11)$$

Hierbei geht der Parameter γ in den variablen Aufwandsanteil für das jeweilige Auf- und Abladen ein und hat die Dimension [ZE/ME], während die Konstante σ für den fixen Aufwandsanteil steht und die Dimension [ZE] hat.

Dabei wird unterstellt, dass die komplette Laubmenge, die aus dem Feld a abgeholt wird, auch komplett vom Transportmittel (Laubwagen) aufgenommen werden kann. Allerdings kann es vorkommen, dass die vom Feld a abzuholende Menge $M(a)$ des Laubhaufens das *maximale Ladevolumen* \bar{T} der Schubkarre überschreitet. In diesem Falle sind mehrere Pendeltouren nötig, um das Feld a vom Laubhaufen zu befreien. Entsprechend ändert sich der Transportaufwand für die Laubentsorgung des Feldes a :

$$TA(a) = \lceil M(a)/\bar{T} \rceil \cdot 2 \cdot D(K, a) \cdot \alpha_T \quad (2.12)$$

bzw.

$$TA(a) = \lceil M(a)/\bar{T} \rceil \cdot 2 \cdot D(K, a) \cdot \alpha_T + \gamma \cdot M(a) + \lceil M(a)/\bar{T} \rceil \cdot \sigma. \quad (2.13)$$

Liegen mehrere Laubhaufen zum Abholen bereit, ergibt sich der gesamte Transportaufwand als die Summe aller Pendeltouren zu allen Laubhaufenfeldern.

Etwas komplizierter wird die Transportaufwandsberechnung dann, wenn zusätzlich zu den Pendeltouren auch Sammeltouren in Frage kommen. Dies hat allerdings nur dann einen Sinn, wenn die auf einer Sammeltour angefahrenen Laubhaufen in ihrer Laubmengensumme das maximale Ladevolumen \bar{T} nicht überschreiten. Im Folgenden wird also davon ausgegangen, dass nur Sammeltouren $K \rightarrow a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_\pi \rightarrow K$ betrachtet werden, für die gilt:

$$\sum_{p=1, \dots, \pi-1} M(a_p) < \bar{T}. \quad (2.14)$$

Hiermit wird sichergestellt, dass bei Weiterfahrt zum nächsten Laubhaufen noch Aufladekapazität in der Schubkarre vorhanden ist. Allerdings wird offen gelassen, ob der letzte Laubhaufen komplett aufgeladen werden kann oder nur ein Teil der Laubmenge $M(a_\pi)$ auf der Schubkarre Platz findet. Dann ergibt sich der Transportaufwand für diese Sammeltour im einfachen Fall, d.h. in Anlehnung an (2.10), zu

$$TA(a_1, \dots, a_\pi) = \alpha_T \cdot [D(K, a_1) + \sum_{p=2, \dots, \pi} D(a_{p-1}, a_p) + D(a_\pi, K)]. \quad (2.15)$$

Bei der komplizierteren Annahme (2.11) statt (2.10) sind dann noch zwei Fälle zu unterscheiden: Die Laubmenge auf Feld a_π passt komplett noch in die Schubkarre oder aber nur ein Teil der Laubmenge $M(a_\pi)$ kann vom letzten Laubhaufen der Sammeltour mitgenommen werden. Entsprechend ergibt sich der Transportaufwand für die Sammeltour zu

$$TA(a_1, \dots, a_\pi) = \alpha_T \cdot [D(K, a_1) + \sum_{p=2, \dots, \pi} D(a_{p-1}, a_p) + D(a_\pi, K)] + \gamma \cdot \sum_{p=1, \dots, \pi} M(a_p) + \pi \cdot \sigma \quad (2.16)$$

bzw.

$$TA(a_1, \dots, a_\pi) = \alpha_T \cdot [D(K, a_1) + \sum_{p=2, \dots, \pi} D(a_{p-1}, a_p) + D(a_\pi, K)] + \gamma \cdot \bar{T} + \pi \cdot \sigma. \quad (2.17)$$

Im Fall (2.16) werden sämtliche auf der Sammeltour angefahrenen Laubhaufenfelder komplett vom Laub befreit, d.h. $M_{neu}(a_p) = 0$ für $p = 1, \dots, \pi$. Im Fall (2.17) gilt dies nur für die Felder a_p , $p = 1, \dots, \pi - 1$, während auf dem letzten Feld a_π zwangsläufig noch Laub verbleibt, und zwar

$$M_{neu}(a_\pi) = M_{alt}(a_\pi) - (\bar{T} - \sum_{p=1, \dots, \pi-1} M_{alt}(a_p)) = \sum_{p=1, \dots, \pi} M_{alt}(a_p) - \bar{T}. \quad (2.18)$$

2.5 Zusammenfassung der Modellgrößen

Es sind bislang schon viele problemrelevante Modellgrößen¹ eingeführt sowie alternative Berechnungsmodelle für den Hark- und Transportaufwand vorgestellt worden. Daher schickt es sich nun an, diese kurz und übersichtlich zusammenzufassen und in ein mathematisches Optimierungsmodell zu überführen.

Ein Garten wird als ein bewerteter Graph $G = (V, E, M, C)$ modelliert, wobei V die Knotenmenge und E die Kantenmenge ist. Die Knoten verkörpern je nach Rasterung rechteckige oder quadratische Felder des Gartens, die für den Hark- und Transportprozess relevant sind. Die Knotenmenge V bestehe aus den Knoten $a_p, p = 1, \dots, P$. Durch die Kanten werden die direkten Nachbarschaften zwischen Feldern beschrieben. Die Knotenbewertungen geben die Laubmengen auf den zugehörigen Feldern an; sie liegen anhand eines P -dimensionalen Vektors M vor. Die Kantenbewertungen geben die Abstände zwischen den Mittelpunkten der benachbarten Felder an; sie liegen anhand einer Matrix C vor. Falls keine Kantenbewertungen angegeben sind, wird durchweg ein Einheitsabstand unterstellt. In diesem Falle wird der Graph auch vereinfacht als ein Tripel $G = (V, E, M)$ geschrieben. Zudem liegt eine Entfernungsmatrix D vor, in der die Distanzen zwischen allen Knoten (Feldern) untereinander und zum Kompostfeld angegeben sind. Diese Matrix D entsteht je nach Vorgabe aus der Matrix C durch Anwendung eines Kürzeste-Wege-Algorithmus² oder durch Berechnung gemäß der Euklidischen Distanz.

In die Berechnung des Hark- bzw. Transportaufwandes gehen die folgenden Modellgrößen³ ein:

¹In diesem Zusammenhang soll kurz auf eine Verfeinerung des Begriffes *Modellgröße* hingewiesen werden, wobei insbesondere auf den Unterschied der Bezeichnungen *Variable* und *Parameter* eingegangen werden soll (vgl. [OvGG09], S. 6): Unter einer *Modellgröße* soll zunächst eine zur Definition der Modellstruktur benötigte Problemgröße verstanden werden (z.B. Laubmenge, Hack- oder Transportaufwandsfaktor). Die Werte einer Modellgröße können als variabel oder als festgesetzt angenommen werden. Als Beispiel verhalte sich eine Ergebnisgröße y in Abhängigkeit von einer Einflussgröße x gemäß dem mathematischen Modell $y = a \cdot x + b$, d.h. es besteht ein linearer Zusammenhang zwischen den Modellgrößen x und y , die als die *Variablen* des Modells bezeichnet werden (unabhängige Variable x , abhängige Variable y). Je nach Setzung der beiden Modellgrößen a und b ($a, b \in \mathbb{R}$) ergibt sich eine bestimmte Gerade im \mathbb{R}^2 . Umgekehrt liefert die beliebige Setzung von a und b die Menge aller Geraden im \mathbb{R}^2 . Man sagt dann auch, dass jede Gerade im \mathbb{R}^2 dem Modell (oder der Modellstruktur) $y = a \cdot x + b$ entspricht und eine spezielle Gerade im \mathbb{R}^2 durch die *Parameter* a und b des Modells spezifiziert wird.

²Hierzu bietet sich das Dijkstra-Verfahren an.

³Hierbei wird $M(a)$ einerseits als Modellparameter (Anfangssetzung) und andererseits als Modellvariable während des Hark- bzw. Transportprozesses benutzt ($M_{alt} \rightarrow M_{neu}$). Eine klare Rollentrennung könnte dadurch herbeigeführt werden, dass ein fest vorgegebener Knotenbewertungsvektor mit M_0 bezeichnet wird und die Variablen $M(a)$ durch $M_0(a)$ initialisiert werden.

Tabelle 2.5: Zusammenfassung der Modellgrößen

Name	Bedeutung	Erläuterung	Dimension
$M(a)$	aktuelle Laubmenge	entspricht zunächst der Laubmenge im Feld a vor dem Harkprozess. Im Verlaufe des Hark- bzw. Transportprozesses verändert sich dieser Wert.	Laubmengen- einheit [ME]
$\bar{M}(a)$	maximale Laubmenge	Der gesamte Harkprozess unterliegt der Nebenbedingung $M(a) \leq \bar{M}(a)$ für alle $a \in V$. Falls diese Obergrenze für alle Felder identisch ist, wird die maximale Laubmenge mit \bar{M} bezeichnet.	Laubmengen- einheit [ME]
\bar{T}	maximales Ladevolumen	entspricht der Ladekapazität des Transporthilfsmittels (Schubkarre)	Laubmengen- einheit [ME]
$\alpha_H(a, b)$	Harkaufwandsfaktor	gibt an, wie viel Zeitkosten beim Harken einer Laubmengeneinheit von einem Feld a zu einem benachbarten Feld b entsteht; falls dieser Aufwand für alle Felderpaare identisch ist, wird der Parameter mit α_H bezeichnet.	Zeitkosteneinheit pro Laubmengen- einheit [ZE/ME]
α_T	Aufwandsfaktor bzgl. Transportmenge	gibt an, wie viel Zeitkosten beim Laubtransport in Abhängigkeit von der Weglänge einer Pendel- oder Sammeltour anfällt.	Zeitkosteneinheit pro Längeneinheit [ZE/LE]
γ	variabler Transportaufwandsfaktor	gibt an, wie viel Zeitkosten beim Aufladen des Laubes in Abhängigkeit von der Laubmenge bei einer Pendel- bzw. Sammeltour anfällt.	Zeitkosteneinheit pro Laubmengen- einheit [ZE/ME]
σ	fixer Transportaufwandsfaktor	gibt an, wie viel Zeitkosten pro Haltepunkt an einem Laubhaufen bei einer Pendel- bzw. Sammeltour anfällt.	Zeitkosteneinheit [ZE]
$HA(a, b)$	Harkaufwand	gibt an, wie viel Zeitkosten für das Harken der Laubmenge vom Feld a in das benachbarte Feld b aufgewendet wird.	Zeitkosteneinheit [ZE]
$TA(a)$	Transportaufwand	gibt an, wie viel Zeitkosten für den Transport der Laubmenge vom Feld a zum Kompost K aufgewendet wird.	Zeitkosteneinheit [ZE]

3 Darstellungsmodell für Harkprozesse und Verfahren zur Harkaufwandsberechnung

Um später zur Entwicklung von effizienten Strategien auf (heuristische und bionische) Verfahren aus der kombinatorischen Optimierung zurückgreifen zu können (Reihenfolge-Permutationen), wird vorweg ein Darstellungsmodell für Harkprozesse eingeführt (Nachfolgerfunktion), an welchem die Berechnung des jeweiligen Harkaufwands vollzogen werden kann.

3.1 Darstellung von Harkprozessen durch Nachfolgerfunktionen

Als eine Lösung des Laubharkproblems gilt eine explizite *Harkvorschrift* in Form einer Abbildung $s : V \rightarrow V$; hierbei wird jedem Feld $a \in V$ dasjenige Nachbarfeld $b = s(a) \in V$ zugewiesen, wohin die Laubmenge $M(a)$ geharkt wird. Eine solche Abbildung wird auch *Nachfolgerfunktion* genannt. Eine anschauliche Darstellung einer Harkvorschrift ist die Tabellenform (vgl. Abb. 2.10):

Tabelle 3.1: Beispiel einer Nachfolgerfunktion

a	1	2	3	4	5	6	7	8	9	10	11	12
$s(a)$	2	6	3	8	9	10	11	12	10	11	11	11

Die obige Nachfolgerfunktion besagt, dass vom Feld 1 zum Feld 2, weiter zum Feld 6, von dort zum Feld 10 und schließlich zum Feld 11 geharkt wird. Wegen $s(11) = 11$ wird im Feld 11 ein Laubhaufen zusammengeharkt; solche Felder werden im Folgenden *Haufenfelder* (oder *Hubs*) genannt. Zudem wird vom Feld 4 zum Feld 8, weiter zum Feld 12 und schließlich zum Laubhaufenfeld 11 geharkt. Die Laubmenge von Feld 5 wird zum Feld 9 weiter zum Feld 10 und schließlich zum Feld 11 geharkt. Letztlich wird noch vom Feld 7 zum Feld 11 geharkt. Das Feld 3 ist wegen $s(3) = 3$ ebenfalls ein Haufenfeld (Hub), allerdings wird diesem Feld kein Beitrag von einem anderen Feld zugeliefert.

Damit eine Abbildung $s : V \rightarrow V$ eine Harkvorschrift darstellt, muss im Zusammenhang mit dem Graph G für jedes $a \in V$ die *Nachbarschaftsbedingung* gelten:

$$a \neq s(a) \Rightarrow [a, s(a)] \in E. \tag{3.1}$$

Hierdurch wird abgesichert, dass im Graph G nur entlang von existierenden Kanten geharkt werden darf. Eine Harkvorschrift in Form einer Nachfolgerfunktion unter Einhaltung der Nachbarschaftsbedingung induziert auf kanonische Weise eine zulässige Lösung des Laubharkproblems, denn es werden die generierten Laubhaufen hinsichtlich der Anzahl, Lage und kumulierten Laubmenge eindeutig festgelegt. Im Folgenden soll durchgängig unterstellt werden, dass eine Harkvorschrift bzw. Nachfolgerfunktion stets der Nachbarschaftsbedingung (3.1) genügt, zumal ein zugehöriges

Prüfmodul sehr einfach implementiert werden kann.

Man mache sich klar, dass durch eine Nachfolgerfunktion s auf kanonische Weise ein Digraph $\vec{G}_s = \langle V, \vec{E}_s \rangle$ induziert wird, wobei die Pfeilmenge \vec{E}_s aus den gerichteten Kanten $\langle a, s(a) \rangle$ mit $a \neq s(a)$ besteht. Dieser Digraph besitzt eine „inverse Baumstruktur“, d.h. der inverse Digraph ist ein gerichteter Wald.¹

An dieser Stelle sei angemerkt, dass durch die Nachfolgerfunktion s zwar einzelne Harkschritte in ihrer Aufeinanderfolge und die obigen „Lösungsparameter“ (Anzahl und Lage der Haufenfelder mit kumulierter Laubmenge) festgelegt werden, aber der gesamte Harkvorgang nicht eindeutig beschrieben wird. So bliebe beispielsweise die Freiheit bestehen, zunächst mit dem gelben Teilprozess zu beginnen, um danach mit dem grünen, dann mit dem roten fortzusetzen und mit dem blauen Teilprozess abzuschließen. Aber auch andere Reihenfolgen sind denkbar. Insbesondere wird nicht unmittelbar festgelegt, ob ein auf kanonische Weise zusammengehöriger Teilprozess auch unterbrochen werden kann, etwa dass der rote Teilprozess zunächst nur bis zum Feld 12, danach der blaue Teilprozess ebenfalls bis zum Feld 12 vollzogen wird, um die dort angesammelte Laubmenge in einem Schritt zum Haufenfeld 13 zu harken. Durch diese verbliebenen Freiheitsgrade ist offensichtlich auch noch Spielraum zur Minimierung des Harkaufwandes vorhanden. Allerdings wird implizit unterstellt und bei den Harkprozeduren im Folgenden auch berücksichtigt, dass in sinnvoller Weise erst dann von einem Feld a aus weitergeharkt wird, wenn sämtliche Vorgängerfelder von a leer geharkt worden sind.²

3.2 Allgemeine Harkaufwandsberechnung für eine gegebene Harkvorschrift

Wegen der Eindeutigkeit der Abbildung s hat jedes Feld a genau einen Nachfolger $s(a)$. Ein Feld a mit $a = s(a)$ stellt ein Haufenfeld (Hub) dar. Zur Bestimmung aller Hubs, die bzgl. eines gegebenen Graphen $G = (V, E)$ durch eine Harkvorschrift bzw. Nachfolgerfunktion s induziert wird, dient der folgende Algorithmus:

Algorithmus 3.1

Setze $HUB(G, s) := \emptyset$.

Für alle $a \in V$ führe aus:

Falls gilt: $a = s(a)$,

¹Im obigen Beispiel ergibt sich ein gerichteter Wald mit zwei Wurzelbäumen, wobei der eine Baum den Wurzelknoten 11 und die Blätter 4, 5, 6 und 7 hat, der andere Baum allein aus dem isolierten Knoten 3 besteht.

²Diese Vorgehensweise entspricht dem Prinzip des sukzessiven Abpflückens eines gerichteten Baumes mit dem Hubknoten als Wurzel bzw. dem inversen Vorgehen bei einer topologischen Sortierung von zyklensfreien Digraphen (vgl. u. a. [Bra94], S. 28f, [HL07], S. 48ff).

setze: $HUB(G, s) := HUB(G, s) \cup \{a\}$.

Die Menge $HUB(G, s)$ stellt die Menge aller Hubs nach komplett ausgeführter Harkprozedur gemäß Harkvorschrift s am Gartenmodell G dar.

Zu jedem Feld a kann auf kanonische Weise die Menge $P(a) = \{b \in V \mid s(b) = a\}$ seiner Vorgänger ermittelt werden.¹ Ein Feld a mit $P(a) = \emptyset$ wird ein *Harkanfangsfeld* oder eine *Harkquelle* genannt.² Ein Haufenfeld a mit $P(a) = \emptyset$ stellt ein *triviales Haufenfeld* dar.

Zur Bestimmung aller Harkquellen dient der folgende Algorithmus:

Algorithmus 3.2

Für alle $a \in V$ führe aus:

Falls gilt: $a \in HUB(G, s)$,

setze: $\delta^-(a) := -1$; andernfalls setze: $\delta^-(a) := 0$.

Für alle $b \in V$ führe aus:

Falls gilt: $s(b) = a$, setze: $\delta^-(a) := \delta^-(a) + 1$.

Die Ergebnisse $\delta^-(a)$ entsprechen den Eingangswerten der Knoten des durch s induzierten Digraphen $\vec{G}_s = \langle V, \vec{E}_s \rangle$. Demnach bedeutet $\delta^-(a) = 0$, dass Knoten a eine Quelle von \vec{G}_s ist.

Für die Berechnung des gesamten Harkaufwandes einer Harkvorschrift ist die Bestimmung derjenigen Felder relevant, von denen aus der jeweils nächste Harkzug gemacht werden soll. Solche Felder sollen als *aktuelle Harkquellen* bezeichnet werden. Diese zeichnen sich im folgenden Algorithmus dadurch aus, dass sie im Verlaufe des Verfahrens den Wert $\delta^-(a) = 0$ annehmen. Dabei bezeichnet $HA(a, s(a))$ eine festgelegte Harkaufwandsberechnung für den Harkzug von Feld a zum Feld $s(a)$ sowie $M(a)$ bzw. $M(s(a))$ die aktuelle Laubmenge im Feld a bzw. $s(a)$.

Algorithmus 3.3

Schritt 1:

Setze: $Anzahl := 0$; $HA_\Sigma(s) := 0$.

Schritt 2:

¹Man beachte, dass die Vorgängermenge von der Harkvorschrift s abhängt, sodass genau genommen $P_s(a)$ als Bezeichnung gewählt werden könnte.

²Der Begriff *Harkquelle* steht in kanonischem Zusammenhang mit dem Begriff *Quelle* von Digraphen: Ein Knoten $a \in V$ eines Digraphen $\vec{G}_s = \langle V, \vec{E}_s \rangle$ heißt eine Quelle, wenn für die Eingangswert $\delta^-(a)$ gilt: $\delta^-(a) = 0$.

Wähle $a \in V$ mit $\delta^-(a) = 0$.

Setze: $Anzahl := Anzahl + 1$.

Falls gilt: $a \neq s(a)$, führe aus:

Berechne $HA(a, s(a))$.

Setze:

$$HA_{\Sigma}(s) := HA_{\Sigma}(s) + HA(a, s(a));$$

$$\delta^-(a) := \delta^-(a) - 1;$$

$$\delta^-(s(a)) := \delta^-(s(a)) - 1;$$

$$M(s(a)) := M(s(a)) + M(a);$$

$$M(a) := 0.$$

Falls gilt: $Anzahl = |V|$, terminiere; andernfalls gehe zu Schritt 2.

Man mache sich klar, dass der obige Algorithmus dafür sorgt, dass von einem Feld a aus erst dann die aktuelle Laubmenge zum Nachfolgerfeld $s(a)$ geharkt wird, wenn sämtliche Vorgängerfelder von a leer geharkt worden sind.¹

Beispielsweise ergibt sich für die obige Nachfolgerfunktion s (vgl. Tab. 3.1) ein Harkaufwand $HA_{\Sigma}(s)$ gemäß (2.6) mit $\alpha_H = 1$ von 47 ZE. Demgegenüber ergibt sich ein Harkaufwand von 124 ZE, falls der gemäß Nachfolgerfunktion s^* induzierte Harkprozess durchgeführt wird (vgl. Tab. 3.2).

Tabelle 3.2: Beispiel einer alternativen Nachfolgerfunktion

a	1	2	3	4	5	6	7	8	9	10	11	12
$s^*(a)$	2	3	4	8	9	5	6	7	10	11	12	12

Beide Harkprozesse erzeugen jeweils *einen* Laubhaufen mit einer kumulierten Laubmenge 29 ME, allerdings an verschiedenen Stellen (Feld 11 bzw. 12). Entsprechend werden sich die jeweils anschließenden Transportprozesse in ihren Transportaufwänden (leicht) unterscheiden. An diesem Beispiel wird deutlich, dass bei festgelegter Berechnungsgrundlage jeder vorgegebenen Nachfolgerfunktion s sowohl der kumulierte Harkaufwand $HA_{\Sigma}(s)$ als auch der kumulierte Transportaufwand $TA_{\Sigma}(s)$ und somit der Gesamtaufwand $GA(s) = HA_{\Sigma}(s) + TA_{\Sigma}(s)$ zugewiesen werden kann. Hierbei seien $HA_{\Sigma}(s)$ und $TA_{\Sigma}(s)$ diejenigen Hark- bzw. Transportaufwände, die sich gemäß den jeweils vereinbarten Rechenvorschriften aus Abschn. 2.3 bzw. 2.4 ergeben.

¹Allerdings bleibt anzumerken, dass bei dieser Harkaufwandberechnung die Zeitkosten für die unproduktiven Wege zwischen den einzelnen Feldern, insbesondere beim Wechsel von Teilharkprozessen, noch unberücksichtigt bleiben. Auf eine entsprechende Verfeinerung der Harkaufwandberechnung durch Einbeziehung dieser (möglicherweise relevanten) Wegaufwände soll hier der Einfachheit halber verzichtet werden.

Modellierung des Laubharkproblems

Wenn man nun die Menge aller möglichen Nachfolgerfunktionen eines gegebenen Graphen $G = (V, E)$ als Lösungsmenge des Laubharkproblems auffasst und diese mit Λ bezeichnet, dann lässt sich das Laubharkproblem allgemein in folgender Form beschreiben:

$$\text{Min! } GA(s) \text{ für } s \in \Lambda \text{ bzw. } \min_{s \in \Lambda} GA(s) \quad (3.2)$$

In folgender Tabelle sind einige verschiedene Nachfolgerfunktionen aufgelistet. Es handelt sich dabei allerdings um solche „Spezialfälle“, bei denen stets *genau ein* Laubhaufen erzeugt wird. Der Transportprozess ist daraufhin ebenfalls speziell organisiert, indem also nur Pendeltouren zwischen Laubhaufen und Kompost zum Tragen kommen; dabei wird die Berechnungsgrundlage (2.10) mit den Distanzen $D(K, a)$ gemäß Manhattan-Metrik bei 1 LE für einen Harkzug und den Parametern $\bar{T} = 10$ und $\alpha_T = 1$ zu Grunde gelegt und der Kompost östlich vom Feld 4 angenommen.

Tabelle 3.3: Ausgewählte Nachfolgerfunktionen mit zugehörigem Gesamtaufwand

a	1	2	3	4	5	6	7	8	9	10	11	12	$HA_{\Sigma}(s_i)$	$TA_{\Sigma}(s_i)$	$GA(s_i)$
$s_1(a)$	2	6	3	8	9	10	11	12	10	11	11	11	47	24	71
$s_2(a)$	2	3	4	8	9	5	6	7	10	11	12	12	124	18	142
$s_3(a)$	2	3	4	4	1	2	3	4	5	6	7	8	76	6	82
$s_4(a)$	2	3	4	4	6	7	8	4	10	11	12	8	76	6	82
$s_5(a)$	2	6	3	8	6	7	7	7	5	11	7	8	46	18	64
$s_6(a)$	5	6	7	8	6	10	11	7	10	10	10	11	46	30	76
$s_7(a)$	5	6	7	8	6	7	8	8	5	6	7	8	59	12	71
$s_8(a)$	2	3	4	4	6	7	8	4	5	6	7	8	76	6	82

Es dürfte unverkennbar sein, dass die Anzahl der verschiedenen Nachfolgerfunktionen, insbesondere bei freier Wahl der Laubhaufenanzahl und einer angepassten Transportstrategie schon bei „kleinen“ Graphen sehr groß ausfällt und mit der Größe des Graphen exponentiell anwächst. Somit besteht eine komplexe Aufgabe im Wesentlichen darin, durch geeignete Hark- und Transportstrategien effiziente Nachfolgerfunktionen zu generieren, um damit das Laubharkproblem zu lösen.

4 Ausblick

In diesem Beitrag werden Möglichkeiten vorgestellt, ein Laubharkproblem als mathematisches Optimierungsproblem zu formulieren, um auf dieser Grundlage geeignete Optimierungsverfahren zur Lösung der Problemstellung entwickeln und anwenden zu können. Dieses erfolgt im nachfolgenden Beitrag dieses Buches (siehe [KSD16]).

Das Laubharkproblem stellt - wie schon zu Beginn erwähnt - ein spezielles Entsorgungsproblem dar. Es wurde deswegen exemplarisch ausgewählt, weil sich jeder Leser und jede Leserin in diese „populäre“ Problemstellung aus eigener Erfahrung oder Anschauung ohne Weiteres eindenken kann. Natürlich lassen sich die vorgestellten Modellierungsansätze auch auf andere Entsorgungsprobleme aus der Praxis adaptieren. Hierbei wird es allerdings darum gehen, die problemspezifischen Besonderheiten in die Modellbildung einzubeziehen.

Literaturverzeichnis

- [Bou02] BOUSONVILLE, T.: Tourenplanung für die Siedlungsabfallentsorgung: Modelle, Methoden und Lösungen zur Optimierung. Wiesbaden, 2002.
- [Bra94] BRANDSTÄDT, A.: Graphen und Algorithmen. Teubner-Verlag, Stuttgart, 1994.
- [Cam96] CAMPBELL, J.F.: Hub location and P-hub median problem. Operations Research, 44:923–935, 1996.
- [HL07] HUSSMANN, S. und LUTZ-WESTPHAL, B. (Herausgeber): Diskrete Mathematik erleben - Anwendungsbasierte und verstehensorientierte Zugänge. Springer Spektrum, Wiesbaden, 2. Auflage, 2007.
- [HP09] HEKMATFAR, M. und PISHAVAAE, M.: Hub location problem. In: FARAHANI, R.Z. und HEKMATFAR, M. (Herausgeber): Facility location: Concepts, models, algorithms and case studies. 2009.
- [JBK13] JANSEN, R., BERKEN, M. und KÖTTER, U.: Handbuch Entsorgungslogistik. Frankfurt a. M., 2013.
- [Kli91] KLINCEWICZ, J.G.: Heuristics for the P-hub location problem. European Journal of Operational Research, 53:25–37, 1991.
- [KSD16] KRUSE, H.-J., SPENST, N. und DERDAU, R.: Heuristische Lösungsstrategien für das Laubharkproblem. In: KRUSE, H.-J. und LASK, T. (Herausgeber): Angewandte mathematische Modellierung und Optimierung – Ausgewählte Modelle, Methoden, Fallstudien, Band 4 der Reihe Forschungsreihe des Fachbereichs

Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Seiten 89–116, 2016. ISSN 2196-6192.

- [OvGG09] ORTLIEB, C.P., v. DRESKY, C., GASSER, I. und GÜNZEL, S.: Mathematische Modellierung. Wiesbaden, 2009.
- [Sou96] SOUREN, R.: Theorie betrieblicher Reduktion: Grundlagen, Modellierung und Optimierungsansätze stofflicher Entsorgungsprozesse. Heidelberg, 1996.
- [Stö93] STÖLZLE, W.: Umweltschutz und Entsorgungslogistik. Berlin, 1993.
- [Wag06] WAGNER, B.: Hub & Spoke-Netzwerke in der Logistik: Modellbasierte Lösungsansätze für ihr Design. Wiesbaden, 2006.
- [WZ98] WIESE, H. und ZELEWSKI, S.: Müllentsorgung und Müllvermeidung: Theoretische und heuristische Ansätze. Zeitschrift für Betriebswirtschaft, (ZfB 1/98):145–176, 1998.

Heuristische Lösungsstrategien für das Laubharkproblem

Hermann-Josef Kruse¹, Natalie Spenst², Ralf Derdau¹

¹Fachbereich Ingenieurwissenschaften und Mathematik, Fachhochschule Bielefeld

²HiQ Projects GmbH, Köln

Kontakt:

hermann-josef.kruse@fh-bielefeld.de

Zusammenfassung: Anhand eines zuvor erstellten mathematischen Modells für ein Laubharkproblem als ein spezielles Entsorgungsproblem werden mehrere Lösungsstrategien entwickelt und im Hinblick auf Effizienz geprüft. Hierbei werden intuitive Vorgehensweisen mit modifizierten Strategien aus dem Bereich der Hub-Location-Problematik sowie für Traveling-Salesman-Probleme vergleichend gegenübergestellt.

Schlüsselwörter: Laubharkproblem, Entsorgungsprozess, heuristische Lösungsverfahren

1 Einleitung

Bei einem realen Laubharkproblem in einem Garten oder Park geht es darum, das auf einer Grundfläche unterschiedlich verteilte Laub zu Haufen zusammenzuharken, diese Laubhaufen auf eine Schubkarre zu laden und zu einer Entsorgungsstelle (Kompost) zu bringen. Durch die Bestimmung einer „optimalen Harkstrategie (wie viele Laubhaufen sollen entstehen und wo platziert werden und von welchen Laubfeldern soll in welcher Reihenfolge zu welchen Haufen geharkt werden) soll im Zusammenhang mit dem daraus resultierenden Aufwand für den Abtransport der Gesamtaufwand (gemessen in Zeit oder Kraft oder Kosten) minimiert werden. Ein solches Laubharkproblem lässt sich als ein diskretes Optimierungsproblem formulieren. Genauer gesagt lässt es sich als ein Spezialfall eines Hub-Location-Problems interpretieren, sodass Verfahren zur Lösung von Hub-Location-Probleme in modifizierter Form auch zur Lösung von Laubharkproblemen eingesetzt werden können.¹

In diesem Beitrag werden verschiedene Lösungsverfahren vorgestellt, die sowohl intuitive Harkstrategien als auch modifizierte Lösungsansätze beinhalten. Insbesondere wird bei den sog. Eröffnungsverfahren auf passende Clusterbildungsstrategien zurückgegriffen, bei den Verbesserungsverfahren werden Anleihen bei bekannten lokalen Suchmethoden gemacht.

Hierzu werden zunächst die grundlegenden Begriffe des mathematischen Optimierungsmodells eines Laubharkproblems in enger Anlehnung an den Beitrag von [Kru16] zusammengefasst. Es werden verschiedene Lösungsverfahren vorgestellt, unterteilt in Eröffnungs- und Verbesserungsverfahren. Abschließend werden Ergebnisse von Effizienzuntersuchungen an diesen Verfahren unterbreitet.

2 Grundlagen des Laubharkproblems

Ein reales Laubharkproblem in einem Garten (oder Park) lässt sich als ein diskretes Optimierungsproblem formulieren², indem zunächst durch Rasterung des Gartens auf kanonische Weise eine $m \times n$ -Matrix G entsteht, in der den einzelnen Feldern ganzzahlige oder reelle Werte als Laubmengen zugeordnet werden oder auch Felder als nicht harkbar (für Teich, Gartenhaus, Bäume) gekennzeichnet werden können (mit -1). Die einzelnen Abstraktionsschritte werden in Abb. 2.1 – Abb. 2.3 skizziert.

¹Zu Hub-Location-Problemen und dessen spezielle Ausprägungen siehe u.a. [Cam96], [HP09], [Kli91], [WZ98].

²Näheres hierzu siehe [Kru16].

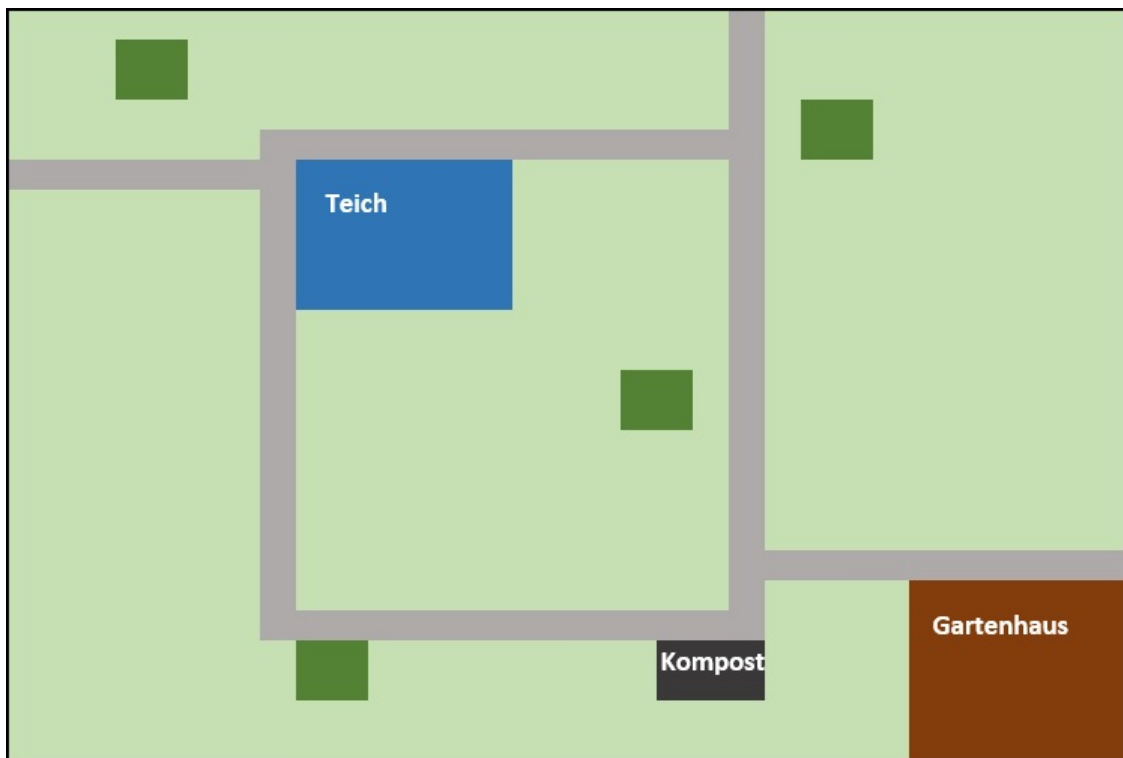


Abbildung 2.1: Skizze eines Gartens (Rasen hellgrün, Bäume grün, Gehwege grau)

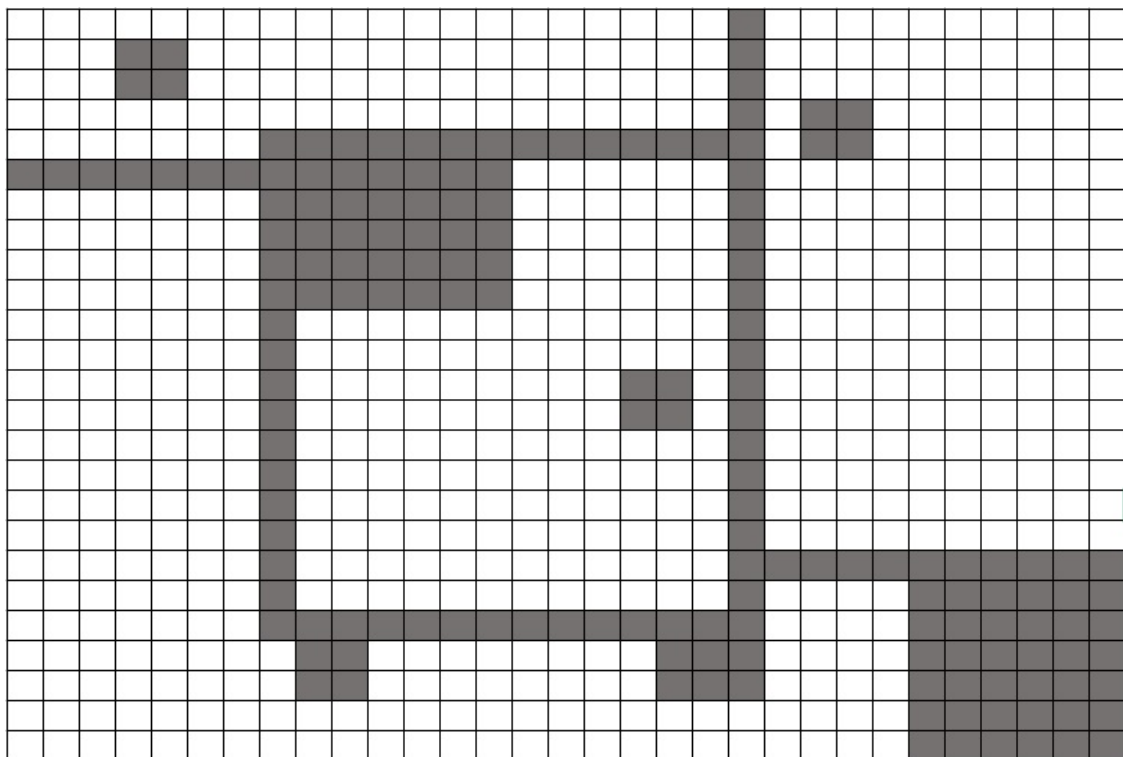


Abbildung 2.2: Rasterung der Gartenskizze

Lösungsstrategien für das Laubharkproblem

8	9	9	9	8	8	7	6	5	4	3	3	3	2	2	1	1	1	1	1	1	1	4	5	6	7	8	7	7	4	3	2
8	9	8	-1	-1	8	8	6	5	5	4	3	3	3	2	1	4	7	7	7	7	7	7	7	7	7	7	7	7	4	3	2
8	9	9	-1	-1	9	9	5	5	5	4	4	3	3	2	2	4	5	6	8	9	9	9	8	8	7	6	5	4	3	2	
7	9	9	8	7	7	7	6	5	4	5	4	4	4	2	2	4	5	7	8	9	8	-1	-1	8	8	6	5	4	3	2	
6	8	8	8	6	6	7	5	4	4	3	3	3	3	3	3	5	5	4	8	9	9	-1	-1	9	9	5	5	4	3	2	
6	7	7	6	5	5	5	5	-1	-1	-1	-1	-1	-1	3	3	4	4	4	7	9	9	8	7	7	7	6	5	5	3	2	
7	6	5	5	3	2	2	3	-1	-1	-1	-1	-1	-1	3	3	4	4	4	6	8	8	8	6	6	7	5	5	4	3	2	
7	5	5	4	3	2	1	2	-1	-1	-1	-1	-1	-1	2	2	2	2	2	6	7	7	6	5	5	5	5	5	4	3	2	
5	5	5	4	3	2	1	1	-1	-1	-1	-1	-1	-1	1	1	4	5	6	7	3	3	3	2	4	4	4	4	4	4	3	
4	4	0	0	0	0	0	1	-1	-1	-1	-1	-1	-1	5	4	7	7	7	7	4	3	3	3	4	4	4	4	4	3	2	
4	4	0	0	0	0	0	0	0	0	0	0	0	0	7	7	9	9	8	8	4	4	3	3	3	2	3	3	3	3	3	
0	0	0	0	0	0	0	0	1	2	2	2	7	7	8	9	9	9	8	8	7	6	4	2	2	2	2	2	2	2	2	
0	0	0	0	0	0	1	1	2	3	3	3	7	7	8	9	8	-1	-1	8	8	6	5	1	0	1	1	1	1	1	1	
0	0	0	0	0	0	1	1	2	4	5	7	7	7	8	9	9	-1	-1	9	9	5	4	1	0	1	1	1	1	1	1	
0	0	0	0	0	0	0	3	3	5	5	4	7	7	7	9	9	8	7	7	7	6	4	1	0	1	1	0	0	0	0	
1	1	0	0	0	0	0	3	3	4	4	4	7	9	6	8	8	8	6	6	7	6	4	1	1	1	0	0	0	0	0	
2	1	1	0	0	0	0	3	3	4	4	4	6	8	6	7	7	6	5	5	5	6	3	1	2	1	1	0	0	0	0	
2	1	0	0	0	0	0	2	2	2	2	2	6	7	4	4	4	4	4	4	3	6	3	1	2	1	0	0	0	0	0	
4	0	0	0	3	2	1	1	1	4	5	6	7	3	4	4	4	4	4	3	2	6	5	1	1	1	0	0	0	0	0	
6	7	7	6	5	5	5	5	4	7	7	7	7	4	3	2	3	3	3	3	3	4	4	3	2	-1	-1	-1	-1	-1	-1	
2	1	3	3	4	7	9	9	9	8	8	7	6	4	1	1	1	1	1	1	1	3	3	3	3	-1	-1	-1	-1	-1	-1	
2	1	4	7	7	7	9	8	-1	-1	8	8	6	4	3	1	1	1	-1	-1	-1	2	2	2	2	-1	-1	-1	-1	-1	-1	
2	2	4	5	6	8	9	9	-1	-1	9	9	5	3	3	1	1	1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	-1	
2	2	4	5	7	8	9	9	8	7	7	7	6	3	3	1	0	0	0	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	
3	3	5	5	4	8	8	8	8	6	6	7	4	2	2	1	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	

Abbildung 2.3: Gartenmatrix

Alternativ ließe sich die Gartenmatrix G auch als ein Graph verstehen, indem die Felder als Knoten und die Nachbarschaft der Felder durch Kanten des Graphen dargestellt werden. Dabei werden nur diejenigen Felder bzw. Knoten als benachbart angesehen, zwischen denen ein direktes Harken der jeweiligen Laubmengen erfolgen kann.¹

Die Felder von G werden entweder mit $G(i, j)$ bezeichnet ($i = 1, \dots, m; j = 1, \dots, n$) oder folgendermaßen durchnummeriert:

$$a = (i - 1) \cdot n + j. \tag{2.1}$$

Die Felder eines Gartenmodells ($m = 3, n = 5$) sind dann wie folgt nummeriert:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Hiernach gilt beispielsweise $G(2, 4) \equiv 9$.

¹Näheres hierzu siehe [Kru16], S. 69.

Als nächstes wird festgelegt, in welcher Art und Weise das Laub verschoben werden kann, also das elementare Harken von einem Feld a in ein benachbartes Feld b geschieht ($a \rightarrow b$). Die Laubmengen $M(a)$ und $M(b)$ von diesen Feldern werden dadurch wie folgt verändert:

$$M_{neu}(b) = M_{alt}(b) + M_{alt}(a), M_{neu}(a) = 0. \quad (2.2)$$

Um diesen Teilprozess durchführen zu können, muss ebenfalls festgelegt werden, was unter der Nachbarschaft von zwei Feldern verstanden wird. In der einfachen Variante sind Felder nur benachbart, wenn sie in der Gartenmatrix G mit einer Seite aneinander grenzen. Zwei Felder a und b sind also Nachbarn ($a \leftrightarrow b$), wenn gilt:

$$a = (i, j) \wedge [b = (i, j - 1) \vee b = (i, j + 1) \vee b = (i - 1, j) \vee b = (i + 1, j)]. \quad (2.3)$$

Damit ist zunächst nur möglich, Laub senkrecht und waagrecht zu harken, also in das nächste obere, untere, rechte oder linke Feld (vgl. Abb. 4 links). Im weiteren Verlauf dieser Ausarbeitung wird jedoch die erweiterte Variante genutzt. Hier ist es auch möglich, das Laub in die diagonal anliegenden Felder zu harken. Somit sind vier weitere Harkmöglichkeiten gegeben (vgl. Abb. 4 rechts):

$$a = (i, j) \wedge [b = (i - 1, j - 1) \vee b = (i - 1, j) \vee b = (i - 1, j + 1) \vee b = (i, j - 1) \vee b = (i, j + 1) \vee b = (i + 1, j - 1) \vee b = (i + 1, j) \vee b = (i + 1, j + 1)]. \quad (2.4)$$

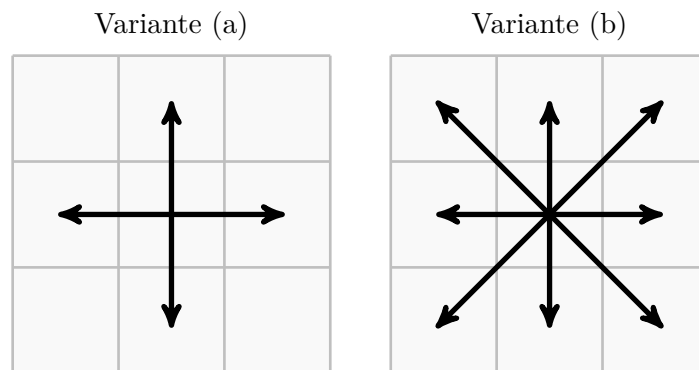


Abbildung 2.4: Nachbarschaftsvarianten

Diese Abstrahierung des Laubharkprozesses als ein diskretes Modell gibt die Möglichkeit, geeignete Entsorgungsstrategien zu entwickeln und zu testen. Die Güte solcher Strategien muss jedoch an einer noch festzulegenden Zielgröße gemessen werden. Die Zielgröße kann die aufgewendete Zeit oder Kraft oder der zurückgelegte Weg sein. Für eine allgemeinere Formulierung wird im Folgenden von „Zeitkosten“ gesprochen, die in der Dimension [ZE] für Zeitkosteneinheit gemessen werden. Es muss sich hierbei nicht unbedingt um Zeit [t] handeln, sondern kann auch als Kosten [GE] verstanden werden. Zu beachten ist, dass im Weiteren alle Teilprozesse des Aufwandsmodells die gleiche Dimension erhalten, damit der gesamte Entsorgungsprozess ein eindeutiges Resultat erzielt.

Lösungsstrategien für das Laubharkproblem

Der gesamte Entsorgungsprozess ist im Wesentlichen in drei Teilprozesse aufzuteilen: Das Harken, das Aufladen und das Abtransportieren des Laubs. Das Harken ist der aufwändigste Teilprozess und wird deshalb in dieser Ausarbeitung besonders ausführlich behandelt. Das Aufladen des Laubs und der Transport zum Kompostfeld werden dagegen einfach gehalten und nur in einer recht einfachen Variante aufgeführt.

Für den Teilprozess des Harkens bieten sich mehrere Aufwandsberechnungen an¹. In diesem Beitrag wird exemplarisch folgende vereinfachte Harkaufwandsberechnung unterlegt:

$$HA(a, b) = \lceil M(a)/M_k \rceil \cdot \alpha_H(a, b). \quad (2.5)$$

Hierbei ist $M(a)$ die Laubmenge im Feld a , die in das benachbarte Feld b geharkt wird, M_k die als Konstante angenommene Laubmenge, die entsprechend der Harke in einem Harkzug bewegt werden kann, und $\lceil \cdot \rceil$ die nächste größere ganze Zahl des Quotienten, falls dieser nicht ganzzahlig ist (Gauß-Klammern). Somit ergibt der *Harkaufwand* $HA(a, b)$ als die mit einem konstanten *Harkparameter* α_H gewichtete Anzahl der Harkzüge, die benötigt werden, um die Laubmenge vom Feld a zum benachbarten Feld b zu harken.

Für den Transportprozess wird die folgende Aufwandsberechnung vorgenommen²:

$$TA(h) = \lceil M(h)/\bar{T} \rceil \cdot 2 \cdot D(K, h) \cdot \alpha_T + \gamma \cdot M(h) + \lceil M(h)/\bar{T} \rceil \cdot \sigma. \quad (2.6)$$

Hierbei ist $M(h)$ die angehäuften Laubmenge eines Haufenfeldes h , \bar{T} das *Ladevolumen* des benutzten Laubwagens, $D(K, h)$ die Euklidische Entfernung zwischen dem Feld h und dem Kompost K , α_T ein konstanter *Transportparameter*, γ ein variabler und σ ein fixer *Aufladeparameter*. Somit enthält $TA(h)$ sowohl den Aufwand für den Transport- als auch für den Aufladeprozess.

Damit ergibt sich der Gesamtaufwand für die komplette Laubentsorgung in einem Garten G auf kanonische Weise als die Summe von einzelnen Hark- und Transportaufwänden.

Zum Erhalt von praxisnahen Lösungen ist es notwendig, dass für einige Parameter gewisse Mindest- bzw. Höchstgrenzen definiert werden. Hier ist unter anderem die *maximale Laubmenge* gefragt, die sich beim Harkvorgang auf einem Feld auftürmen lässt und die es schon aus rein natürlichen Stabilitätsgründen gibt. Solch eine Laubmenge kann für jedes Feld individuell bestimmt werden, d.h. $M(a) \leq \bar{M}(a)$, oder als gemeinsame Obergrenze für alle Felder festgelegt werden, d.h. $M(a) \leq \bar{M}$. Im Folgenden wird nur noch der einfache Fall einer gemeinsamen maximalen Laubmenge \bar{M} angenommen. In diesem Zusammenhang stellt sich ebenfalls die Frage, wie viele dieser Laubhaufen

¹Näheres hierzu siehe [Kru16], S. 71f.

²Näheres hierzu siehe [Kru16], S. 78ff.

aufgehäuft werden sollen. Hier existiert als theoretische Obergrenze die Anzahl der Felder, d.h. es könnte in jedem Feld ein Laubhaufen liegen, was aber die Frage nach dem Zusammenharken erübrigt. Die mindestens benötigte Anzahl an Laubhaufen ist hier eine Restriktion, die für die Berechnungsverfahren entscheidend ist. Um diese Untergrenze zu bestimmen, muss die *gesamte Laubmenge* M_{ges} , d.h. wie viel Laub sich insgesamt in der Gartenmatrix G befindet, ins Verhältnis zur maximalen Laubmenge \overline{M} gestellt werden. So kann dann die *Laubhaufenuntergrenze* H_{UG} durch den aufgerundeten Quotienten von gesamter Laubmenge M_{ges} und maximaler Laubmenge \overline{M} gebildet werden:

$$H_{UG} = \lceil M_{ges}/\overline{M} \rceil. \quad (2.7)$$

3 Harkstrategien als Eröffnungsverfahren

Zunächst werden zwei Eröffnungsverfahren vorgestellt, die jeweils das Clustering als Grundlage aufweisen. Die beiden Eröffnungsverfahren unterscheiden sich allerdings in der Art und Weise, wie und wann die umliegenden Felder dem ausgewählten Hub h zugeordnet werden. Beim *statischen Verfahren* werden im ersten Schritt Cluster gebildet, für die im zweiten Schritt Hubs gesucht werden, denen dann die übrigen Felder zugewiesen werden. Das *dynamische Verfahren* bildet erst im Laufe der Zuordnungen die Cluster. Hier wird zunächst ein Hub ermittelt und diesem dann nach einer bestimmten Vorschrift Felder zugeordnet. Danach wird der nächste Hub gesucht.

Bei beiden Verfahren werden die zur Beschreibung der Harkvorschrift gebildeten *Nachfolgerfunktionen* s in einer Matrix abgespeichert. Jedes zugeordnete Feld a hat dann einen Nachfolger $s(a)$. Hieraus kann dann pro Cluster eine eindeutige Harkvorschrift abgelesen werden.¹

3.1 Erstes Eröffnungsverfahren: Statische Harkstrategie

Das erste Eröffnungsverfahren ahmt das intuitive Verhalten einer Person beim Laubharken nach: Schritt 1: Einteilung des Gartens in eine geeignete Anzahl etwa gleich großer Abschnitte (*Clusterbildung*).

Schritt 2: In jedem Abschnitt wird ein Feld h als Hub festgelegt (*Hub-Bestimmung*).

Schritt 3: Das zu einem Abschnitt gehörige Laub wird auf den jeweiligen Hub geharkt (*Festlegung der Harkvorschrift*).

Das nachfolgende Verfahren orientiert sich an diesen drei Schritten, berücksichtigt zudem aber auch die durch die maximale Laubmenge \overline{M} vorgegebene Begrenzung der Größe der Cluster. Bei der Bestimmung eines Hubs h für ein Cluster C wird $M(h) = \max_{a \in C} M(a)$ berücksichtigt.

¹Näheres hierzu siehe [Kru16], S. 82f.

Lösungsstrategien für das Laubharkproblem

Im ersten Schritt dieses Verfahrens wird der Garten durch ein simples Ablaufen der Felder in Zickzack-Form in Cluster eingeteilt, welche die maximale Laubmenge \bar{M} nicht überschreiten. Von der ersten zur letzten Zeile wird eine Spalte abgelaufen und alle betretenen Felder in einen Cluster eingefügt, bis die maximale Laubmenge \bar{M} erreicht ist. Ist die letzte Zeile einer Spalte erreicht und \bar{M} noch nicht erreicht, findet ein Wechsel in die nächste Spalte statt, wobei die Zeilen nun von der letzten zur ersten abgelaufen werden. So wird verhindert, dass bei geringer Maximallaubmenge nicht-zusammenhängende Cluster entstehen. Das Prinzip dieses Clusterings wird am Beispiel einer 6×10 -Matrix deutlich (Abb. 3.1).

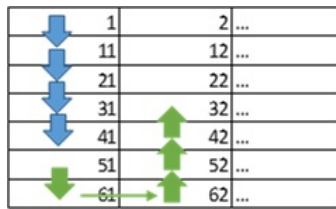


Abbildung 3.1: Clusterbildung durch Zickzack-Prozess

Ein Problem tritt auf, wenn die Felder durch ein Ausschlussfeld (-1) getrennt werden und nicht auf anderem Weg miteinander verbunden werden können. Abbildung 3.2 zeigt so ein Beispiel, in dem der erste Cluster aus fünf Feldern besteht, die aber durch das Ausschlussfeld in der Mitte voneinander getrennt werden. Das Laub der anderen Felder kann nicht über dieses Feld geharkt werden, was verhindert, dass das gesamte Laub dieses Clusters auf einem Hub zusammengeführt werden kann.

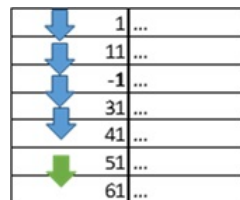


Abbildung 3.2: Problem bei nicht-harkbaren Feldern

Dieser Spezialfall kann verhindert werden, indem der aktuelle Cluster geschlossen wird und ein neuer Cluster begonnen wird, wenn ein Ausschlussfeld betreten wird. Damit befindet sich so ein Ausschlussfeld immer als erstes Feld in einem Cluster und kann andere Felder innerhalb des Clusters nicht voneinander trennen. Diese Problemumgehung kann jedoch dazu führen, dass auch Cluster mit nur sehr wenigen Feldern, z.B. auch nur mit einem, gebildet werden. Umgekehrt können in einer großen Gartenmatrix überwiegend einspaltige Cluster entstehen. In diesem Falle müsste das Zickzack-Prinzip an passenden Teilmatrizen vollzogen werden.

Nachdem Cluster gebildet und alle Felder der Gartenmatrix zugeordnet sind, wird im zweiten Schritt für jeden Cluster C_i das Feld a mit der bereits größten Laubmengen $M(a) = \max_{b \in C_i} M(b)$ bestimmt.¹ Das gefundene Feld ist damit der Hub h_i des Clusters C_i .

¹Ist dieses Feld nicht eindeutig bestimmbar, wird ein sekundäres Auswahlkriterium benutzt (z.B. per Zufall oder nach FIFO-Prinzip).

Im dritten Schritt werden nun alle anderen Felder des Clusters auf kürzestem Weg zu dem Hub geharkt. Hierzu wird zunächst die Menge B_h der direkten Nachbarfelder des Hubs h gesucht und mit den Feldern im Cluster verglichen. Diejenigen Felder a , die sowohl Hub-Nachbarn als auch Cluster-Element sind, werden auf den Hub geharkt (siehe Abb. 3.3 (a)).

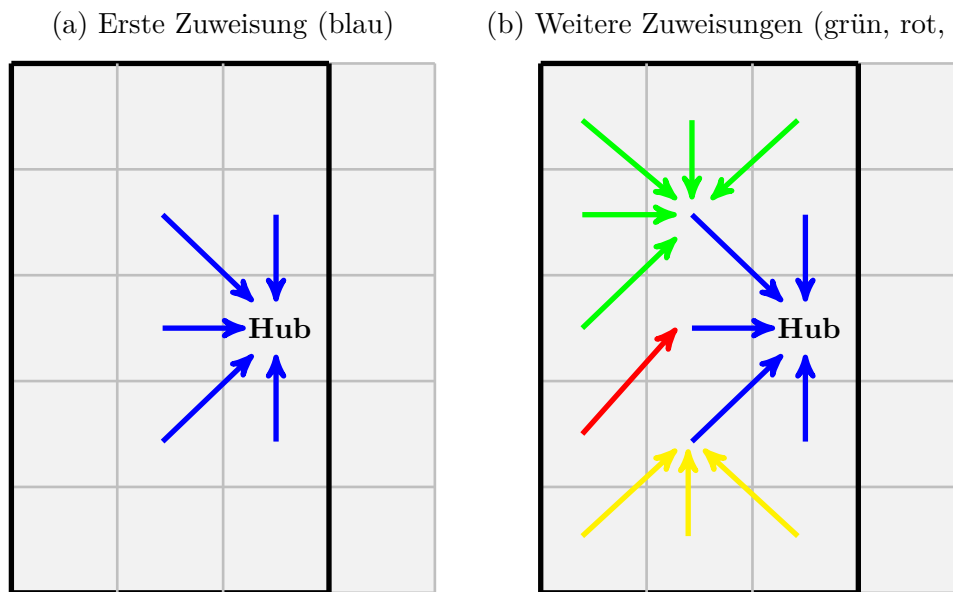


Abbildung 3.3: Feldzuweisungen bei der statischen Harkstrategie innerhalb eines Clusters (schwarz umrandet)

Nun werden zu den geharkten Feldern ebenfalls direkte, noch nicht geharkte Nachbarn gesucht, die sich zusätzlich noch innerhalb des betrachteten Clusters befinden. Hier wird mit dem zuerst gefundenen und zugeordneten Nachbarn begonnen.¹

In der nächsten Iteration wird überprüft, ob es noch Felder innerhalb des Clusters gibt, die noch nicht zugeordnet sind. Ist das der Fall, werden wieder, beginnend beim ersten zugeordneten Feld, die direkten Nachbarn gesucht. Sind alle Felder zugeordnet, wechselt das Verfahren in den nächsten Cluster (siehe Abb. 3.3 (b)).

3.2 Zweites Eröffnungsverfahren: Dynamische Harkstrategie

Das zweite Eröffnungsverfahren ist bei der Clusterbildung wesentlich dynamischer. Es unterscheidet sich vom obigen auch in der Reihenfolge der Teilprozessschritte. Im Gegensatz zu dem zuvor beschriebenen statischen Verfahren werden hier zunächst keine einschränkenden Cluster gebildet. Diese ergeben sich erst durch die Nachfolgersuche und die Beschränkung durch die maximale Laubmenge \bar{M} .

¹Im implementierten Verfahren werden Schlangenspeicher (FIFO) zu Grunde gelegt; diese könnten aber auch durch Stapelspeicher (LIFO) ersetzt werden.

Jede Iteration i dieses Verfahrens beginnt mit der Suche nach einem (weiteren) Hub h_i , indem ein Feld mit der größten Laubmenge unter allen noch nicht einem Cluster zugeordneten Felder von G bestimmt wird. Ist ein Hub h_i bestimmt, wird das Feld für die weitere Hub-Suche ausgeschlossen und seine Laubmenge als *aktuelle Harklaubmenge* M_h des Hubs h_i gespeichert. Mit diesem Hub-Feld geht das Verfahren in die Nachfolgersuche. Hierbei werden ausgehend von einem angegebenen Feld auf rekursive Weise Nachbarfelder gesucht, die diesem unter bestimmten Bedingungen zugeordnet werden. Wird also die Nachfolgersuche mit einem Feld a gestartet, werden zu diesem Feld zuerst einmal alle Nachbarn, d.h. alle Felder aus der Nachbarmenge B_a bestimmt. Diese Nachbarfelder werden dann auf folgende Eigenschaften geprüft:

1. Das Nachbarfeld $b \in B_a$ besitzt noch keinen Nachfolger $s(b)$, ist also noch keinem Cluster zugeordnet.
2. Die Obergrenze \bar{M} wird nicht überschritten, wenn die Laubmenge $M(b)$ des Nachbarfeldes $b \in B_a$ der aktuellen Harklaubmenge M_h bzw. das Feld b dem Cluster C_i hinzugefügt wird.
3. Die Laubmenge $M(b)$ dieses Nachbarfeldes ist nicht größer als die Laubmenge $M(a)$ des betrachteten Feldes a , also $M(b) \leq M(a)$.

Die erste Bedingung stellt sicher, dass jedes Feld nur einen Nachfolger erhält. Die Harkfolge eines Feldes muss eindeutig sein. Das zweite Prüfkriterium garantiert, dass die Harklaubmenge innerhalb eines Clusters die maximale Laubmenge nicht überschreitet (wie beim vorigen Verfahren). Das dritte Kriterium hat den Hintergrund, dass der Harkaufwand minimiert werden soll. Mit jedem zur Harkfolge hinzugefügten Feld muss das Laub dieses Feldes zusätzlich geharkt werden. Nimmt die Laubmenge der Felder nun sukzessiv ab (oder bleibt zumindest gleich), ist die Zunahme des Harkaufwandes geringer als bei erlaubter Hinzunahme von größer werdenden Laubmengen. Es wird also verhindert, dass große Laubmengen über beliebige Felder zum Hub geharkt werden. Falls nun alle drei Kriterien auf das Nachbarfeld b zutreffen, wird es in die *Menge der potenziellen Nachfolger*, B_{pot} , aufgenommen. Welches Nachbarfeld b nun das aktuelle Feld a als Nachfolger erhält (d.h. $s(b) = a$), entscheidet wieder deren Laubmenge. Nur dasjenige Nachbarfeld mit der kleinsten Laubmenge wird schließlich in den Cluster aufgenommen und erhält das betrachtete Feld a als Nachfolger. Dieses Vorgehen verhindert, zusammen mit dem dritten Kriterium im Auswahlverfahren, dass lange, sich schlängelnde Harkfolgen entstehen. Es werden vielmehr kurze Harkfolgen gebildet. Nachdem das Feld aufgenommen und seine Laubmenge zur aktuellen Harklaubmenge der Iteration hinzuaddiert wurde, wird nun erneut die Nachfolgersuche aufgerufen, diesmal mit dem soeben hinzugefügten Nachbarfeld b . Wird kein weiterer Nachfolger gefunden, finden systematische Rückwärtsschritte (*Backtracking*) statt. Hierbei wird das jeweils aktuelle Feld wiederholt der Nachfolgersuche unterzogen. Somit ergibt sich auf kanonische Weise eine Baumstruktur (gerichteter Baum mit dem Hub als Wurzel), durch welche die Harkvorschrift beschrieben wird.

Es folgt ein Beispiel für einen solchen Zuordnungsprozess (siehe Abb. 3.4). Das aktuell betrachtete Feld ist jeweils schwarz umrandet.

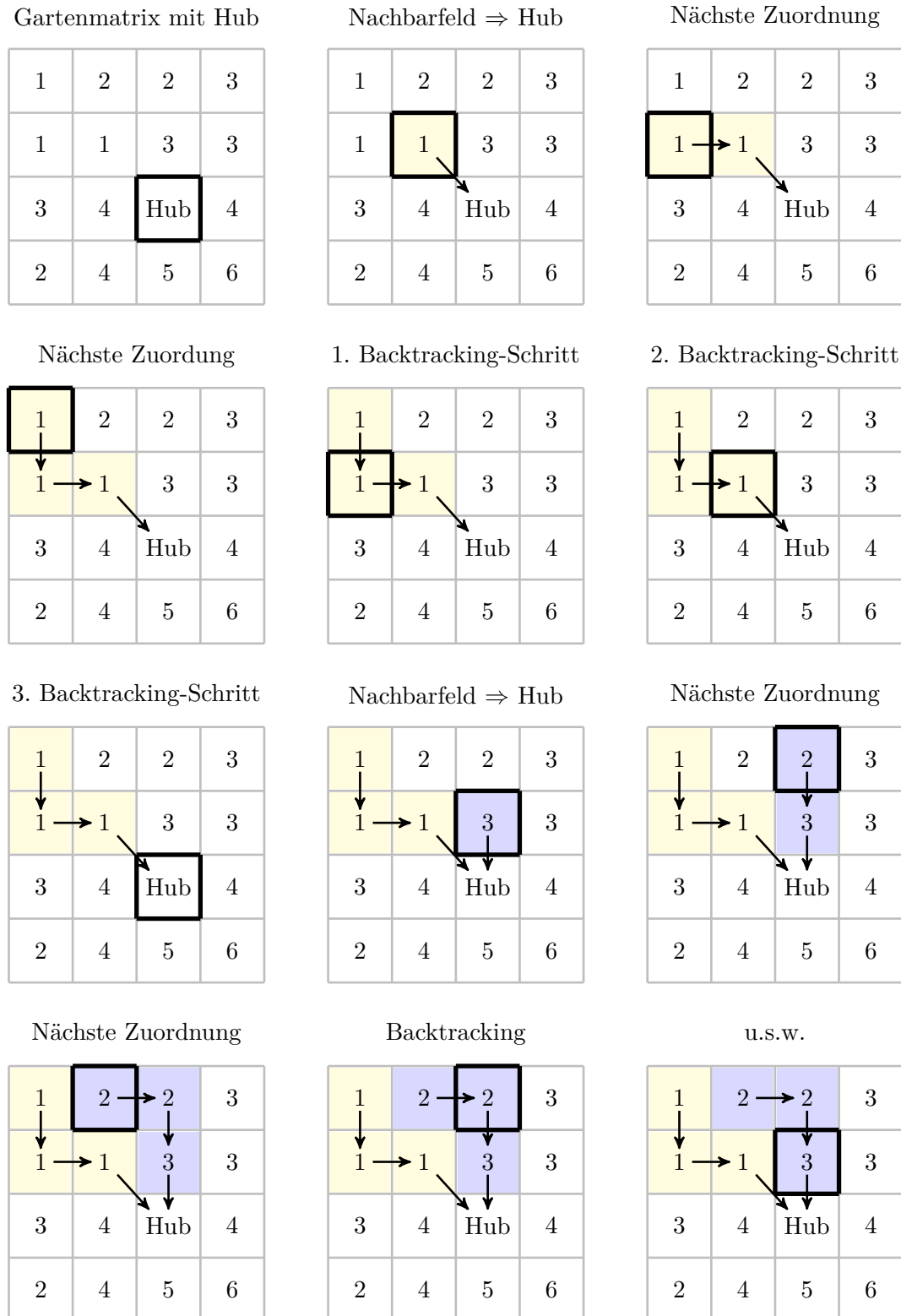


Abbildung 3.4: Zuordnungsprozess bei dynamischer Harkstrategie

Die Suche wird erst abgebrochen, wenn der Hub keine potenziellen Nachfolger mehr findet oder die maximale Laubmenge \overline{M} erreicht ist. Trifft eines zu, wird die Suchfunktion beendet und geprüft, ob noch Felder existieren, die keinem anderen Feld zugeordnet sind. Unter diesen Feldern wird dann der nächste Hub gesucht und somit die nächste Iteration gestartet. Bei dieser Art von Nachbarzuordnung kann es allerdings vorkommen, dass ein Feld als Hub bestimmt wird, dem aber keine anderen Felder zugeordnet werden.

4 Verbesserungsverfahren

Die zwei vorgestellten Eröffnungsverfahren bilden nach unterschiedlichen Vorgehensweisen erste Lösungsmöglichkeiten für das Laubharkproblem, die alle Kapazitätsrestriktionen berücksichtigen. Um diese Zulässigkeit der Lösungen bei Verbesserungsversuchen zu erhalten, müssen bei den Verbesserungsverfahren gewisse Einschränkungen beachtet werden. Insbesondere muss garantiert werden, dass es sich bei einer geänderten Nachfolgerfunktion wieder um eine Harkvorschrift handelt. Eine Nachfolgerfunktion s bildet eine Harkvorschrift, wenn folgende Merkmale auf sie zutreffen:

1. Pro Cluster wird die maximale Laubmenge \overline{M} nicht überschritten.
2. Es ist genau ein Hub pro Cluster vorhanden.¹
3. Der Nachfolger jedes Feldes ist ein direkter Nachbar des Feldes.
4. Jedes Feld wird nur zu genau einem Nachfolger geharkt.
5. Die Harkfolgen bilden keine Kreisläufe.²

Für jede Bedingung muss also eine Prüfung stattfinden, die innerhalb des Optimierungsverfahrens darauf prüft, ob die entstandene Harkvorschrift noch zulässig ist. Diese Prüfungen werden nacheinander ausgeführt. Bei einem Verstoß gegen eine Zulässigkeitsbedingung wird der Prüfprozess abgebrochen und zum Verbesserungsverfahren zurückgekehrt.

4.1 Lösungsverbesserung durch Nachfolgertausch innerhalb der Cluster

Dieses Verbesserungsverfahren tauscht innerhalb eines jeden Clusters C_i für jedes Feld a seinen Nachfolger $s(a)$ durch ein anderes Feld $b \in C_i$ aus und ermittelt dabei, ob die sich daraus ergebende Lösung besser ist als die bestehende. Es handelt sich also um ein relativ einfaches Tauschverfahren, das für jeden Cluster separat durchgeführt wird. Im Laufe des Verfahrens wird unterschieden, ob es sich beim aktuell betrachteten Feld um einen Hub handelt oder um ein einfaches Feld. Ist

¹Durch dieses Kriterium wird impliziert, dass kein Cluster innerhalb eines Clusters entstehen darf. Durch Verzicht auf diese Einschränkung entstehen womöglich interessante Verfahrenserweiterungen, die hier allerdings nicht weiter untersucht werden.

²Grundsätzlich könnte eine durch einen Verbesserungsschritt entstehende Harkfolge einen Kreislauf bilden und somit nicht mehr der kanonischen Konvention entsprechen, dass alle Harkfolgen in einem Hub enden. Näheres hierzu siehe [Spe15].

das betrachtete Feld ein Hub, muss bei dem Tausch des Nachfolgers darauf geachtet werden, dass weiterhin ein Hub in dem Cluster existiert. Diese Bedingung wird dadurch gelöst, dass dasjenige Feld, welches der neue Nachfolger des Hubs wird, selber die Rolle des Hubs übernimmt. In beiden Fällen wird nach dem Tausch eine Prüfung der Harkvorschrift durchgeführt. Werden die Regeln einer Harkvorschrift verletzt, wird der Tausch rückgängig gemacht und das nächste Feld wird auf Tauschbarkeit geprüft. Liegt mit dem Tausch jedoch weiterhin eine Harkvorschrift vor, kann diese Konstellation der Güteprüfung unterzogen werden, d.h. ob dieser Nachfolgertausch eine Verringerung des Harkaufwandes bewirkt hat. Trifft dies nicht zu, werden die Änderungen wieder rückgängig gemacht. Ansonsten werden die neuen Nachfolger übernommen und der aktuelle Harkaufwand auf den verbesserten Wert gesetzt.

Algorithmus 4.1

Gegeben:

Nachfolgerfunktion s , zugehöriger Harkaufwand $HA(s)$ und *ein* durch s induzierter Cluster C von G (o.B.d.A. sei die Mächtigkeit von C größer 1)

Setze: $HA_{aktuell} := HA(s)$.

Für alle $a \in C$ führe aus:

Falls $a \neq s(a)$, führe aus:

Für alle $b \in C$ mit $b \neq a$ führe aus:

Setze: $temp := s(a)$, $s(a) := b$.

Starte Harkvorschriftsprüfung für s .

Falls $Check = FALSE$, setze: $s(a) := temp$;

andernfalls führe aus:

Berechne $HA(s)$.

Falls $HA(s) \geq HA_{aktuell}$, setze: $s(a) := temp$;

andernfalls setze: $HA_{aktuell} := HA(s)$.

andernfalls (d.h. $a = s(a)$) führe aus:

Für alle $b \in C$ mit $b \neq a$ führe aus:

Setze: $temp := s(b)$, $s(a) := b$, $s(b) := a$.

Starte Harkvorschriftsprüfung für s .

Falls $Check = FALSE$, setze: $s(a) := a$, $s(b) := temp$;

andernfalls führe aus:

Berechne $HA(s)$.

Falls $HA(s) \geq HA_{aktuell}$,

setze: $s(a) := a, s(b) := temp$;
 andernfalls setze: $HA_{aktuell} := HA(s)$.

Ergebnis: Neue Nachfolgerfunktion s mit verbessertem Harkaufwand im selben Cluster C .

4.2 Lösungsverbesserung durch Einfügestrategie

Das zweite Verbesserungsverfahren ist nicht auf die einzelnen Cluster eingegrenzt, sondern sucht für jedes Feld geeignete Nachfolger in der nächsten Nachbarschaft des Feldes und tauscht diese ein. Für jedes Feld $a \in G$ werden zunächst die Nachbarn $b \in B_a$ bestimmt. Nun werden nacheinander diese Nachbarn als Nachfolger $s(a)$ des Feldes a eingesetzt. Dabei findet wieder die Überprüfung auf Gültigkeit der Harkvorschrift und ggf. eine Güteprüfung statt. Ist der Harkaufwand geringer, wird die aktuelle Harkvorschrift übernommen und in der nächsten Iteration verwendet. Verbildlicht passiert der folgende Schritt:

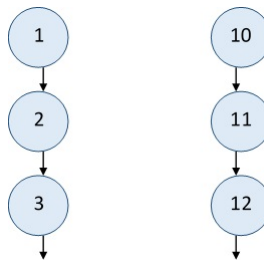


Abbildung 4.1: Ausgangslage

In dem oberen Beispiel sind zwei Harkfolgen zu sehen, die zu verschiedenen Clustern gehören. Das Verfahren ersetzt nun den Nachfolger eines Feldes durch einen anderen, wenn die Harkvorschrift erhalten bleibt und der Harkaufwand dadurch geringer wird. Die Vorgänger dieses Feldes bleiben seine Vorgänger, werden also in die neue Harkfolge mitgenommen. In diesem Beispiel bekommt das Feld 2 als neuen Nachfolger das Feld 11 (bzw. es wird das Feld 2 nun Vorgänger des Feldes 11).

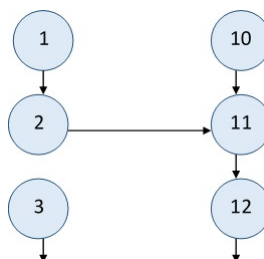


Abbildung 4.2: Neue Clusterbildung

Die Harkfolge ändert sich somit dahingehend, dass das Feld 11 nun zwei Vorgänger hat, das Feld 3 hat keine Vorgänger mehr. Die Nachbarschaftsbedingung und die $n-1$ -Zuweisung bleiben weiterhin bestehen. Diese Art des Harkfolgentausches basiert auf dem Relocate-Verfahren von Savelsbergh

(1992)¹. Diese Heuristik wurde entwickelt, um in der Tourenplanung entwickelte Routen zu verbessern. Hierbei wird ein Knoten aus einer Route entnommen und in eine andere Route eingefügt. Dieses Verfahren wurde von Van Breedam (2001)² dahingehend modifiziert, dass nicht nur ein einzelner Knoten zwischen zwei Touren getauscht wird, sondern eine Kette von Knoten. Dieses Prinzip wird in abgewandelter Form in diesem Verbesserungsverfahren angewandt. Obwohl nur ein Feld in eine andere Harkfolge eingefügt wird, zieht dieses Feld seine eigenen Vorgängerfelder mit in die Harkfolge hinein. Hier ist es, anders als beim Tourenplanungsproblem, nicht notwendig, dass die eingefügte Harkfolge genau zwischen zwei Feldern eingefügt wird, da eine Harkfolge keine 1-1-Verbindung ist, sondern viele Quellknoten besitzen darf. Somit wird durch das Einfügen des Feldes einfach eine neue Harkquelle in den bestehenden Cluster eingefügt.

Algorithmus 4.2

Gegeben: Nachfolgerfunktion s , zugehöriger Harkaufwand $HA(s)$.

Setze: $HA_{aktuell} := HA(s)$.

Für alle $a \in G$ mit $s(a) \neq a \wedge s(a) \neq -1$ führe aus:

Ermittle Nachbarmenge B_a .

Für alle $b \in B_a$ mit $s(b) \neq -1$ führe aus:

Setze: $temp := s(a)$, $s(a) := b$.

Starte Harkvorschriftsprüfung für s .

Falls $Check = FALSE$, setze: $s(a) := temp$;

andernfalls führe aus:

Berechne $HA(s)$.

Falls $HA(s) \geq HA_{aktuell}$, setze: $s(a) := temp$;

andernfalls setze: $HA_{aktuell} := HA(s)$.

Ergebnis: Neue Nachfolgerfunktion s mit verbessertem Harkaufwand bei ggf. neuen Clustern.

5 Beispielrechnungen und Bewertung der Verfahren

Die Effizienzuntersuchungen an den vorgestellten Verfahren sowohl im Einzelnen als auch in Kombination miteinander und insbesondere in Abhängigkeit von den Parametereinstellungen sind gegenwärtig noch nicht abgeschlossen. Daher kann die Lösungsgüte der Verfahren hier auch nur sehr allgemein angegeben werden. Zunächst werden die vier vorgestellten Verfahren im Folgenden

¹Vgl. [Sav92].

²Vgl. [VB01].

Lösungsstrategien für das Laubharkproblem

auf zwei ausgewählte Beispielmatrizen angewendet, um einen Gütevergleich tendenziell aufzuzeigen. Abschließend werden allgemeinere Effizienzaussagen gemacht, die sich aus größeren Testreihen tendenziell schließen lassen.

5.1 Beispielrechnungen

Das erste für die Beispieltests genutzte Gartenmodell ist eine zufällig erzeugte 10×20 -Matrix. Diese Matrix G_1 ist mit insgesamt 932 ME Laub gefüllt und besitzt 7 Ausschlussfelder. Zu den Ausschlussfeldern gehört auch der Kompost **K**, der die Koordinaten (10,18) hat.

6	4	2	5	7	4	7	3	5	0	5	6	5	3	5	5	4	3	0	9
2	3	5	2	4	2	2	6	6	3	4	6	6	8	6	6	5	5	6	9
8	-1	4	3	7	7	-1	4	3	5	3	5	8	9	8	-1	2	4	7	5
3	4	2	4	2	4	6	4	6	5	3	3	3	1	5	4	3	7	6	4
12	2	1	4	3	5	4	5	6	6	6	2	9	2	6	6	4	13	9	4
7	3	2	6	5	6	6	-1	5	7	-1	3	2	3	4	5	7	5	5	2
6	2	8	7	2	4	6	5	4	10	5	4	8	3	8	3	3	3	2	1
6	8	6	7	4	3	7	5	8	2	4	3	10	6	3	5	3	7	9	2
4	3	-1	4	4	4	4	3	7	9	3	8	4	5	1	3	5	6	4	2
6	9	1	7	6	6	5	6	5	4	0	4	4	5	5	6	6	K	5	10

Das zweite Gartenmodell G_2 ist einem praxisnahen Modell nachempfunden. Die Größe dieses Modells entspricht wieder einer 10×20 -Matrix. Es hat mehrere Ausschlussflächen, die Gebäude darstellen sollen, und separate Ausschlussfelder, die Bäume darstellen. Außerdem liegt das Laub vermehrt um die Bäume, da es ja von ihnen abfällt. Die gesamte Laubmenge M_{ges} beträgt ebenfalls 932 ME und auch der Kompost befindet sich weiterhin an der für Modell G_1 genannten Position.

3	2	0	3	3	2	4	3	3	2	1	2	3	4	0	3	2	6	-1	-1
4	7	3	2	3	2	6	5	4	5	6	6	5	5	8	9	6	13	-1	-1
5	11	12	13	5	4	6	6	5	7	6	5	4	4	10	12	14	12	7	6
3	6	-1	9	6	4	2	2	1	-1	-1	-1	-1	8	9	11	-1	10	8	7
2	5	4	4	4	5	4	3	1	-1	-1	-1	-1	8	9	12	11	12	8	7
1	4	3	4	3	13	14	12	2	-1	-1	-1	-1	9	8	6	5	6	7	7
3	8	11	13	9	13	-1	10	10	9	3	4	4	8	-1	5	1	-1	-1	-1
4	4	-1	10	9	9	12	9	6	8	7	1	3	5	5	1	1	-1	-1	-1
5	7	8	11	8	5	6	7	7	6	6	2	3	3	3	3	1	-1	-1	-1
1	2	5	4	2	1	1	2	5	4	5	1	2	3	2	2	1	K	-1	-1

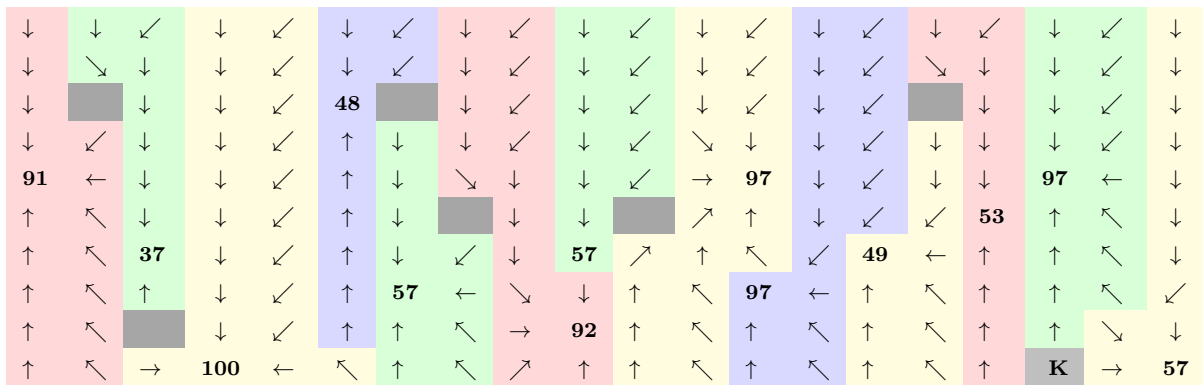
Die maximale Laubmenge \bar{M} sei für beide Gartenmodelle auf 100 ME gesetzt, identisch mit dem maximalen Ladevolumen der Schubkarre. Die übrigen Parameter sind zunächst auf den Wert 1

gesetzt, um die Ergebnisse der Verfahren nicht zu verzerren. In der folgenden Übersicht sind die Parametereinstellungen abzulesen:

Tabelle 5.1: Parametereinstellungen

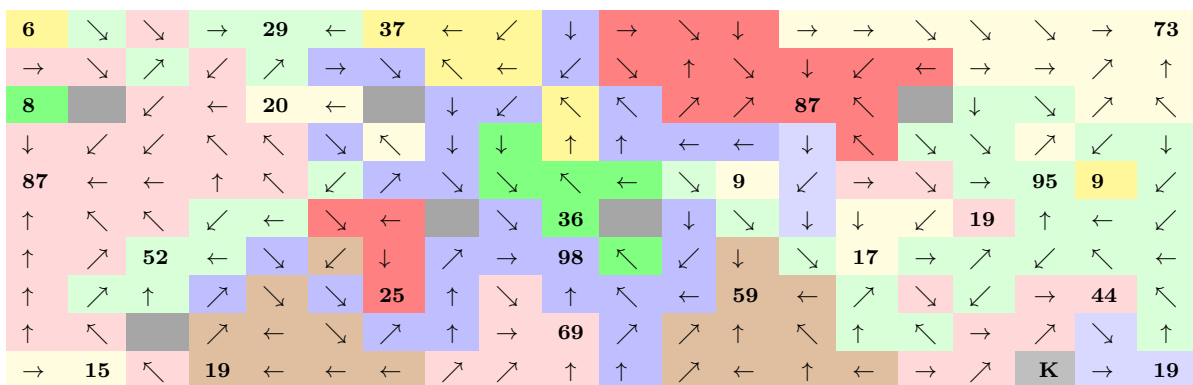
Parameter	Wert
Maximale Laubmenge \bar{M}	100
Maximales Ladevolumen \bar{T}	100
Kritische Laubharkmenge M_k	1
Harkparameter α_H	1
Transportparameter α_T	1
Variabler Aufladeparameter γ	1
Fixer Aufladeparameter σ	1

Zuerst werden beide Eröffnungsverfahren an der zufällig erzeugten Gartenmatrix G_1 getestet. Das statische Eröffnungsverfahren erzeugt dabei folgende Harkvorschrift:



Der Harkaufwand beträgt bei dieser Lösung 2676 ZE, der Transportaufwand beträgt 1178,04 ZE. Somit kommt dieses Verfahren auf einen Gesamtaufwand von 3854,04 ZE.

Das dynamische Eröffnungsverfahren erzeugt aus der Gartenmatrix G_1 folgende Harkvorschrift:

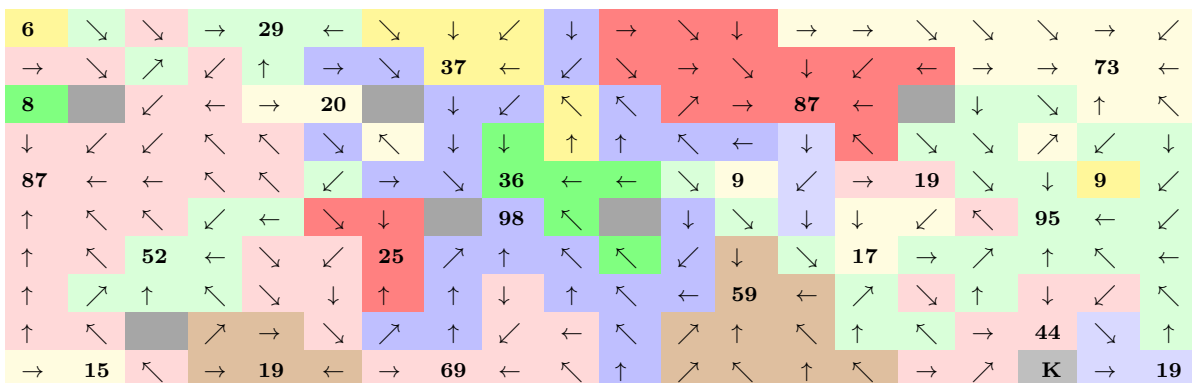


Lösungsstrategien für das Laubharkproblem

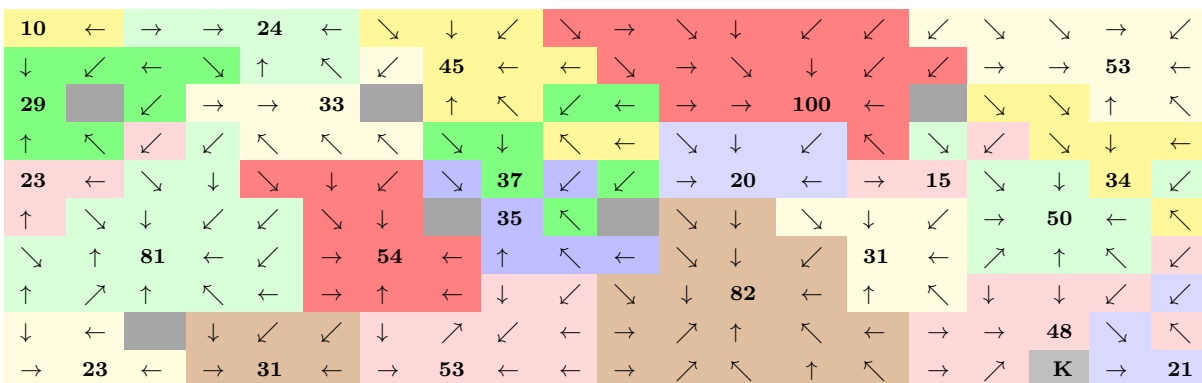
Der Harkaufwand beträgt hier nur 1774 ZE, was vornehmlich auf die höhere Anzahl an Hubs zurückzuführen ist. Der Transportaufwand ist dadurch jedoch höher und beträgt 1424,19 ZE, womit der Gesamtaufwand also bei 3198,19 ZE liegt.

Diese Startlösungen können nun jeweils mit den vorgestellten Verbesserungsverfahren optimiert werden. Im Folgenden soll aber nur die bessere der beiden Lösungen als Startlösung gewählt werden. Entsprechend wird für das Modell G_1 das Ergebnis des dynamischen Verfahrens als Ausgangslösung dienen. Zur Verminderung des Gesamtaufwands werden die Verbesserungsverfahren in folgender Reihenfolge auf die Ausgangslösungen angewendet: Zuerst die Verbesserung innerhalb der Cluster, dann das Verbesserungsverfahren über alle Cluster und zum Schluss noch einmal der Nachfolgertausch innerhalb jeden Clusters.¹

Das erste Verbesserungsverfahren, das Tauschen von Nachfolgern innerhalb der einzelnen Cluster, verringert den Gesamtaufwand auf 2849 ZE, wobei der Harkaufwand auf 1426 ZE vermindert wird, der Transportaufwand nur auf 1423 ZE. Es ergibt sich die folgende Harkvorschrift:

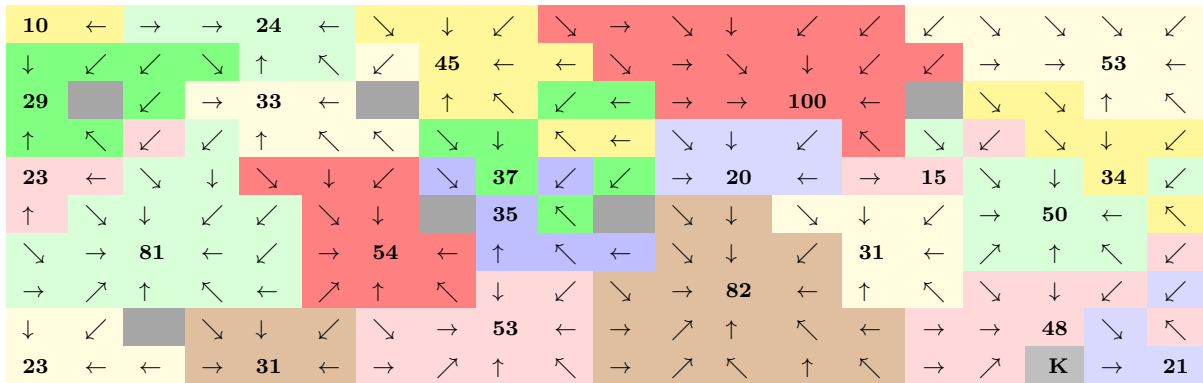


Das zweite Verbesserungsverfahren, das Einfügen von Feldern in benachbarte Harkfolgen, kann diese Lösung nochmals verbessern. Mit einem Harkaufwand von 1076 ZE und gleichbleibendem Transportaufwand ergibt sich ein Gesamtaufwand von 2499 ZE.



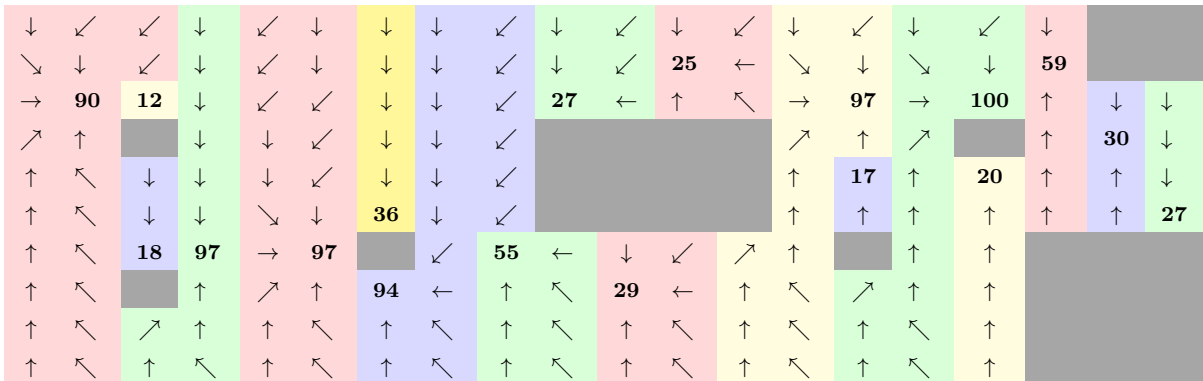
¹Diese Vorgehensweise wird hier exemplarisch vorgestellt, weil sie sich aus den bisherigen Testreihen als sehr vorteilhaft erwiesen hat.

Durch das nochmalige Ausführen des ersten Verbesserungsverfahrens kann der Gesamtaufwand sogar noch einmal vermindert werden (auf 2430,86 ZE). Der Harkaufwand beträgt jetzt nur noch 1006 ZE, der Transportaufwand hat sich jedoch wieder ein wenig erhöht (auf 1424,86 ZE).



Insgesamt konnte der Gesamtaufwand beim Gartenmodell G_1 um ca. 24 % gegenüber der gewählten Startlösung verbessert werden.

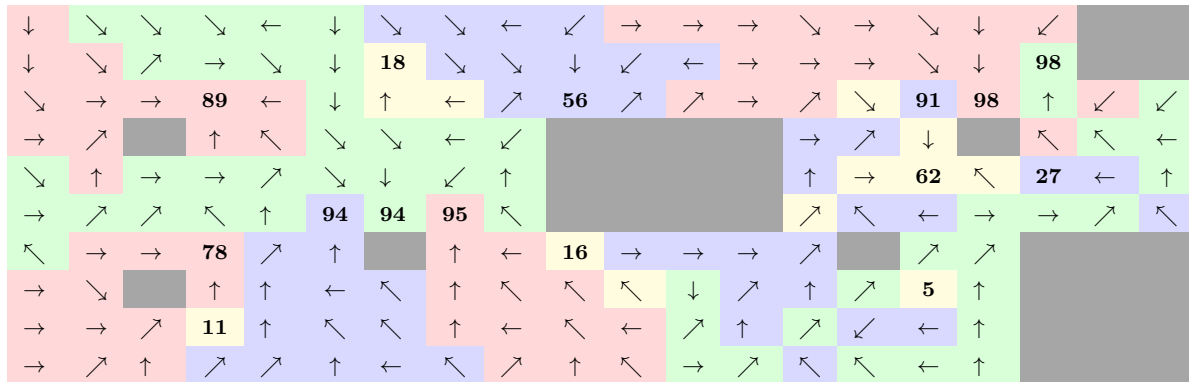
Als nächstes werden die beiden Eröffnungsverfahren auf das praxisnahe Modell G_2 angewendet. Mit dem statischen Eröffnungsverfahren ergibt sich die folgende Harkvorschrift :



Der Harkaufwand beläuft sich auf 1789 ZE, der Transportaufwand auf 1310,91 ZE, woraus sich der Gesamtaufwand 3099,91 ZE ergibt.

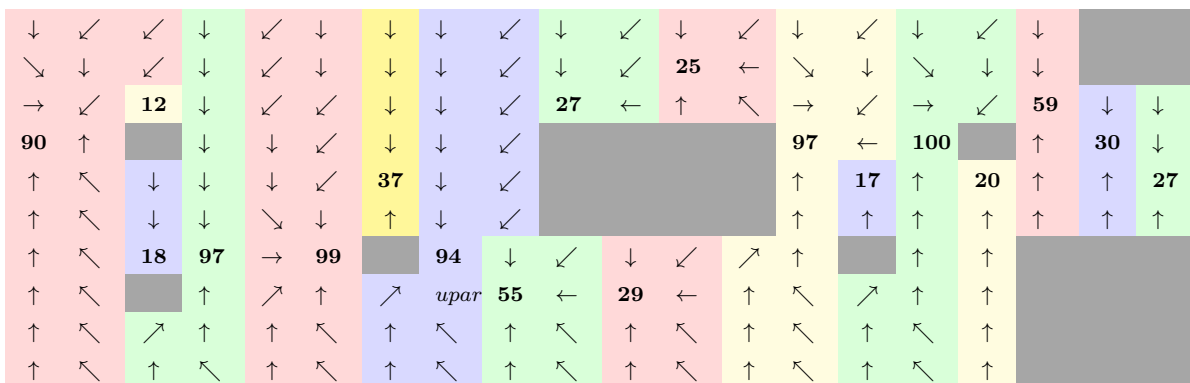
Mit dem dynamischen Verfahren ergibt sich aus dem Gartenmodell G_2 folgende Harkvorschrift:

Lösungsstrategien für das Laubharkproblem



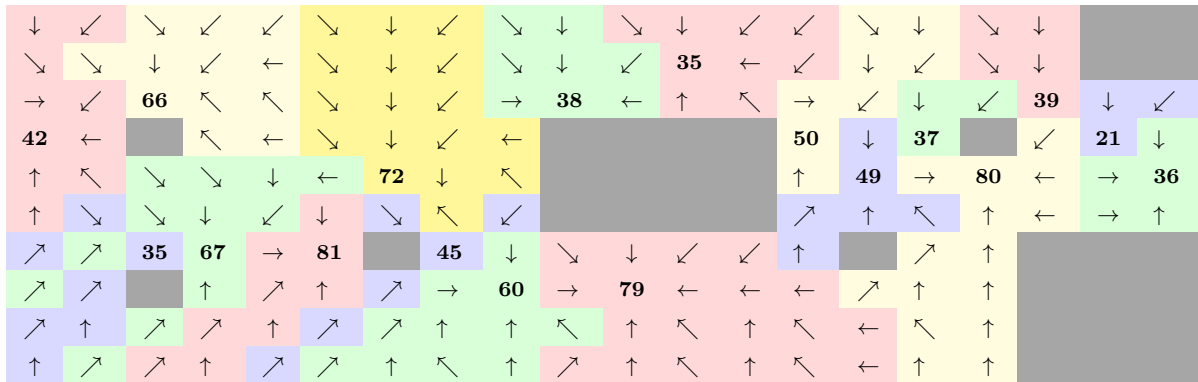
Hier beträgt der Harkaufwand nun 2333 ZE, also wesentlich mehr als beim statischen Verfahren; der Transportaufwand liegt jedoch nur bei 1241,94 ZE. Insgesamt berechnet dieses Verfahren einen Gesamtaufwand von 3574,94 ZE.

Die bessere Ausgangslösung für Modell G_2 ist somit die Lösung des statischen Eröffnungsverfahrens, die nun als Startlösung dient. Auch hier wird nun zuerst das Verbesserungsverfahren innerhalb der Cluster angewendet.



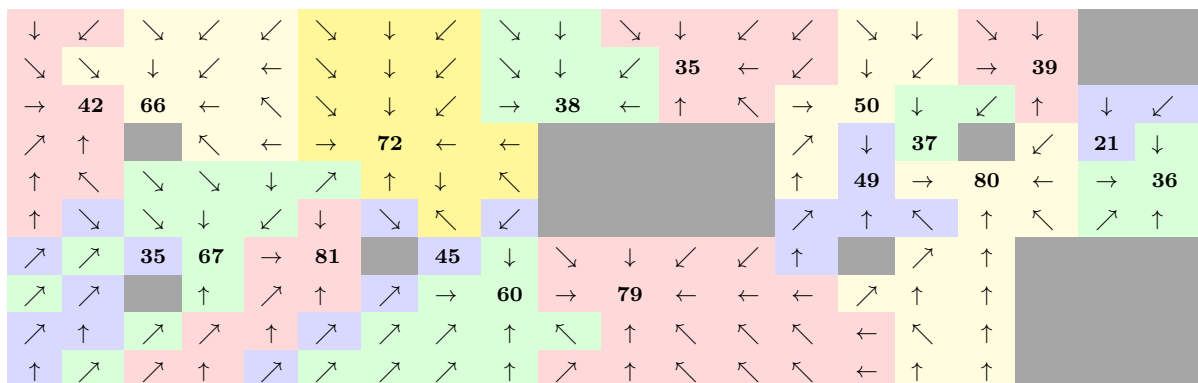
Der Harkaufwand hat sich von 1789 ZE auf 1658 ZE verbessert, der Transportaufwand nur sehr gering von 1310,91 ZE auf 1308,30 ZE. Der Gesamtaufwand in Höhe von 2966,30 ZE hat sich bei diesem Verbesserungsverfahren demnach recht wenig verändert (um 4,31%).

Nachdem nun darauf das zweite Verbesserungsverfahren durchgeführt wurde, ändert sich die Harkvorschrift wie folgt:



Der Gesamtaufwand fällt nun deutlich geringer aus (2533,30 ZE). Der Harkaufwand hat sich dabei um ca. 400 ZE auf 1225 ZE verringert.

Der zweite Durchlauf des ersten Verbesserungsverfahrens verbessert diese Lösung nun stärker. Bei leicht erhöhtem Transportaufwand mit 1310,88 ZE verringern sich der Harkaufwand auf 1109 ZE und der Gesamtaufwand somit auf 2419,88 ZE.



Insgesamt konnte der Gesamtaufwand um ca. 22% verbessert werden.

5.2 Weitere Testergebnisse

Die an zwei ausgewählten Beispielen aufgezeigten Verhaltensweisen der vorgestellten Verfahren (Abschn. 5.1) lassen sich anhand von weiteren Tests tendenziell bestätigen. Insbesondere erweist sich eine kombinierte Vorgehensweise mit beiden Verbesserungsverfahren als besonders effizient.

Im Folgenden sollen aber auch zu weiteren Verfahrenseigenschaften Stellung bezogen werden. Hierzu werden verschieden große Gartenmodelle erzeugt und die Ergebnisse des Hark-, Transport- und Gesamtaufwandes anhand der einzelnen Verfahren vergleichend gegenübergestellt. Die Größe der Gärten – gemessen an der Anzahl der Felder – erstreckt sich dabei von 36 (klein) bis 2000 (mittelgroß).

Lösungsstrategien für das Laubharkproblem

Zunächst sollen die reinen und kombinierten Verbesserungsverfahren auch im Zusammenhang mit der Wahl der Ausgangslösung verglichen werden. Dafür werden die folgenden Abkürzungen vereinbart:

- $S \equiv$ Statisches Eröffnungsverfahren,
- $D \equiv$ Dynamisches Eröffnungsverfahren,
- $N \equiv$ 1. Verbesserungsverfahren („Nachfolgertausch“),
- $R \equiv$ 2. Verbesserungsverfahren („Relocate“).

Somit bedeutet $S_N_R_N_R$, dass das Ergebnis des statischen Eröffnungsverfahrens als Ausgangslösung genommen und die sich anschließenden Verbesserungsschritte im jeweiligen Wechsel zunächst mit dem 1., dann mit dem 2. Verbesserungsverfahren u.s.w. vorgenommen worden sind. Entsprechend bedeutet D_R4 , dass die Startlösung aus dem dynamischen Eröffnungsverfahren mit 4 Iterationen des 2. Verbesserungsverfahrens behandelt worden ist.

Hierbei zeigen sich für alle Gartengrößen zwei deutliche Tendenzen: Zum einen erweisen sich die kombinierten gegenüber den reinen Verbesserungsverfahren in nahezu allen Fällen als die besseren Lösungsansätze. Zum anderen zeigt sich die Auswahl der Eröffnungsstrategie als entscheidungsrelevant für die Lösungsgüte. Dies soll exemplarisch am Beispiel eines Gartenmodells mit der Größe 400 demonstriert werden, wobei die Zahlen 1 bis 5 die Anzahl der Iterationen angibt (1 \equiv Ergebnis nach Eröffnungsverfahren, 2 \equiv Ergebnis nach anschließender Iteration mit Verbesserungsverfahren, u.s.w.):

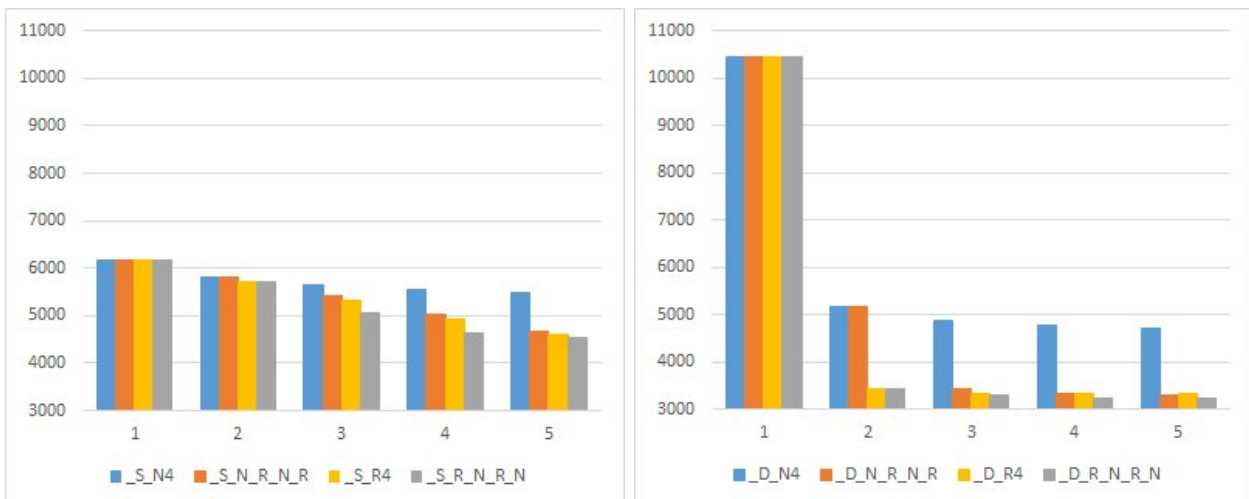


Abbildung 5.1: Vergleich von Verfahrensreihenfolgen mit statischer Eröffnung (links) und dynamischer Eröffnung (rechts)

Offensichtlich liefert das statische Eröffnungsverfahren eine signifikant bessere Startlösung, welche in den folgenden Verbesserungsschritten moderat verbessert wird. Demgegenüber schaffen die Verbesserungsverfahren zum Teil erhebliche Verbesserungen der eindeutig schlechteren Startlösung durch das dynamische Eröffnungsverfahren. Allerdings führen die kombinierten Verfahren in fast

allen untersuchten Fällen zu besseren Lösungen. Demgegenüber ergibt sich keine klare Dominanz der Reihenfolgen $*_N_R_$ und $*_R_N_$ für den direkten Anschluss an ein Eröffnungsverfahren.

In den folgenden Abbildungen werden die relativen Verbesserungen gegenüber den jeweiligen Startlösungen (100%) durch kombinierte Verbesserungsschritte an unterschiedlich großen Gärten demonstriert: In beiden Fällen liefert das reine Verbesserungsverfahren N keine signifikanten Veränderungen nach der ersten Iteration, wohingegen das Verbesserungsverfahren R auch in der reinen Form (Rx) mit den kombinierten Vorgehensweisen „Schritt halten“ kann ($x \equiv$ Anzahl der Iterationen).

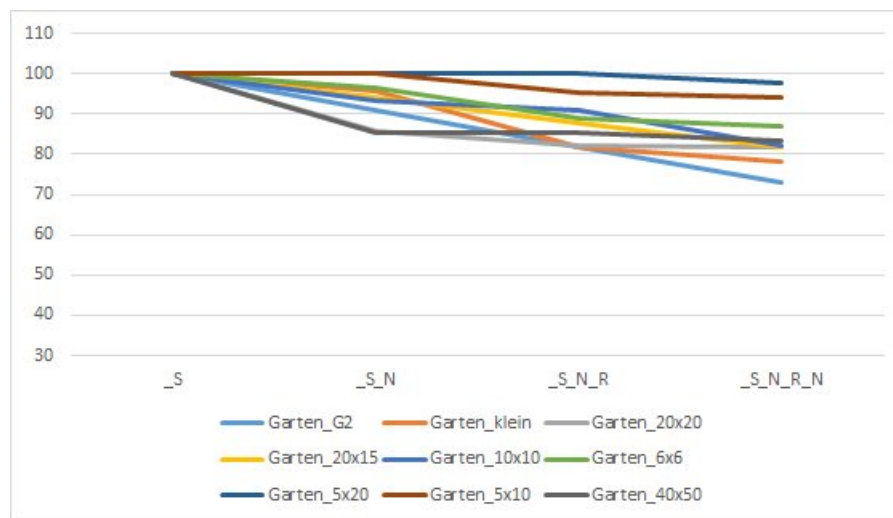


Abbildung 5.2: Relative Verbesserung des Gesamtaufwands bei statischem Eröffnungsverfahren

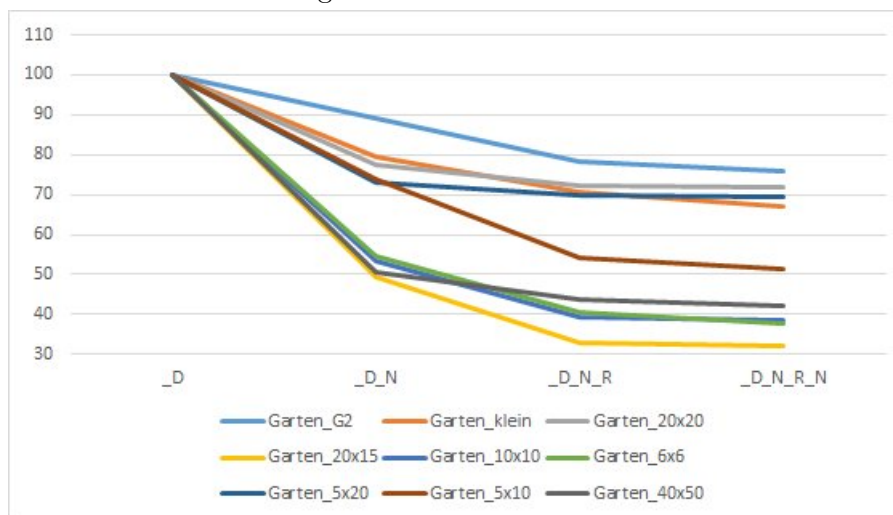


Abbildung 5.3: Relative Verbesserung des Gesamtaufwands bei dynamischem Eröffnungsverfahren

Zum einen zeigt sich auch hier das bereits oben angesprochene Phänomen, dass die Verbesserungsverfahren sehr unterschiedlich auf die beiden sehr different strukturierten Eröffnungslösungen reagieren. Zum anderen wird deutlich, dass die Verbesserungen bereits nach wenigen Iterationen

Lösungsstrategien für das Laubharkproblem

stark abnehmen und somit bereits von einer stabilen Lösung ausgegangen werden kann. Während in der obigen Darstellung beide Startlösungen jeweils auf 100% gesetzt sind, zeigt die nächste Abbildung auch den Unterschied der Startlösungen, indem jeweils nur die Lösung des statischen Eröffnungsverfahrens auf 100% gesetzt ist.

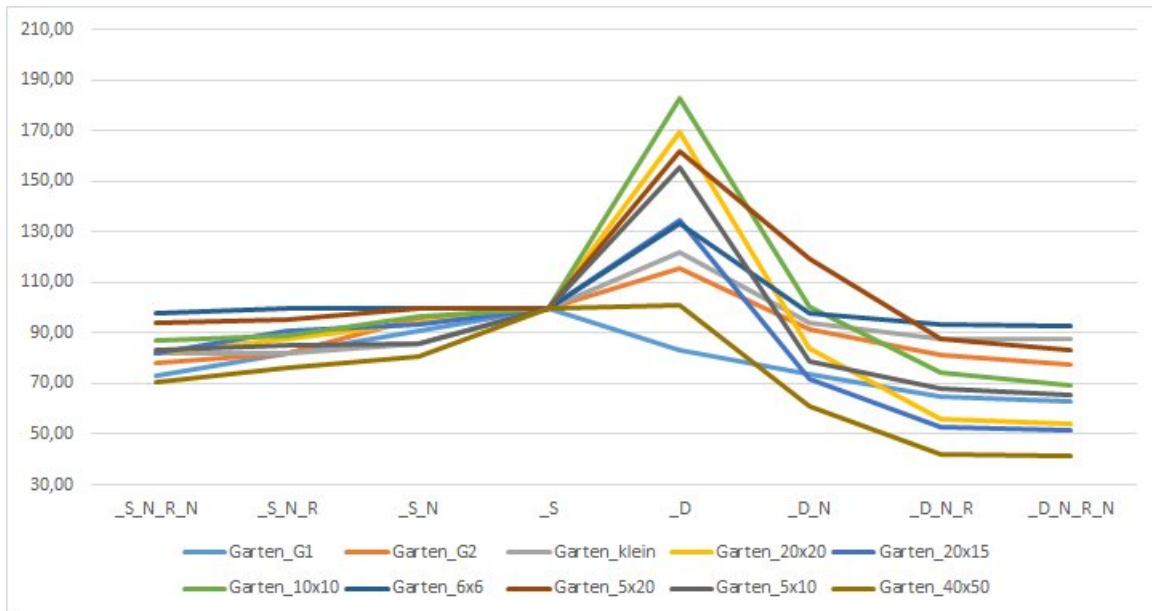


Abbildung 5.4: Gesamtaufwand bezogen auf das statische Eröffnungsverfahren bei Fortsetzung mit Reihenfolge _N_R_

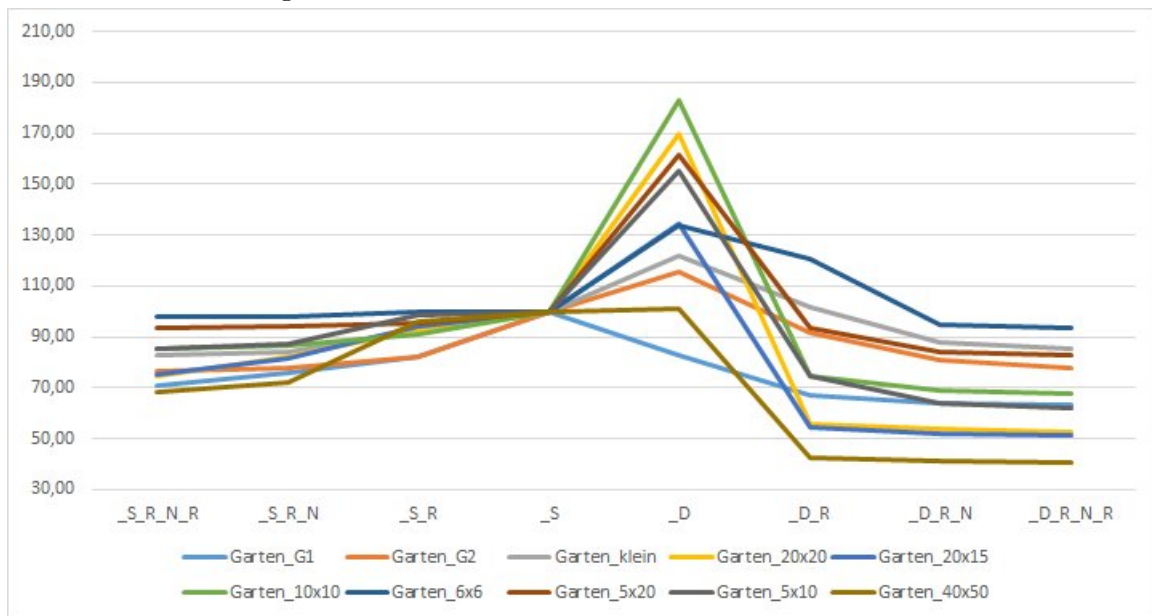


Abbildung 5.5: Gesamtaufwand bezogen auf das statische Eröffnungsverfahren bei Fortsetzung mit Reihenfolge _R_N_

Hierbei lässt sich kein signifikanter Unterschied zwischen den Reihenfolgen _N_R_ und _R_N_ als Fortsetzung des statischen oder des dynamischen Eröffnungsverfahrens erkennen.

Interessant ist auch zu beobachten, wie unterschiedlich sich die Anteile von Hark- und Transportaufwand auf den Gesamtaufwand in Abhängigkeit der Eröffnungsstrategie auswirken (siehe Abschn. 5.1). Dieser Sachverhalt lässt sich sehr einfach durch die unterschiedlichen Clusterbildungen erklären, an denen nur das 2. Verbesserungsverfahren per Konstruktion noch Änderungen bewirken können (hier am Beispiel eines Gartens der Größe 20×20).



Abbildung 5.6: Anteile von Hark- und Transportaufwand am Gesamtaufwand

Von großer Bedeutung für die Anwendung der Verbesserungsverfahren überhaupt (ob in reiner oder kombinierter Form) ist deren Rechenzeitverhalten, zumal der Zeitaufwand für beide Eröffnungsverfahren als unkritisch angesehen werden kann. Es zeigt sich, dass das „Nachbarschaftsverfahren“ (N) stets zeitaufwändiger als das „Reloaded-Verfahren“ (R) ist, sodass der Zeitaufwand für die gemischten Strategien zwischen denen der reinen Strategien liegt. Dies sei exemplarisch wieder am Beispiel eines Gartenmodells mit der Größe 400 (20×20) dargestellt:

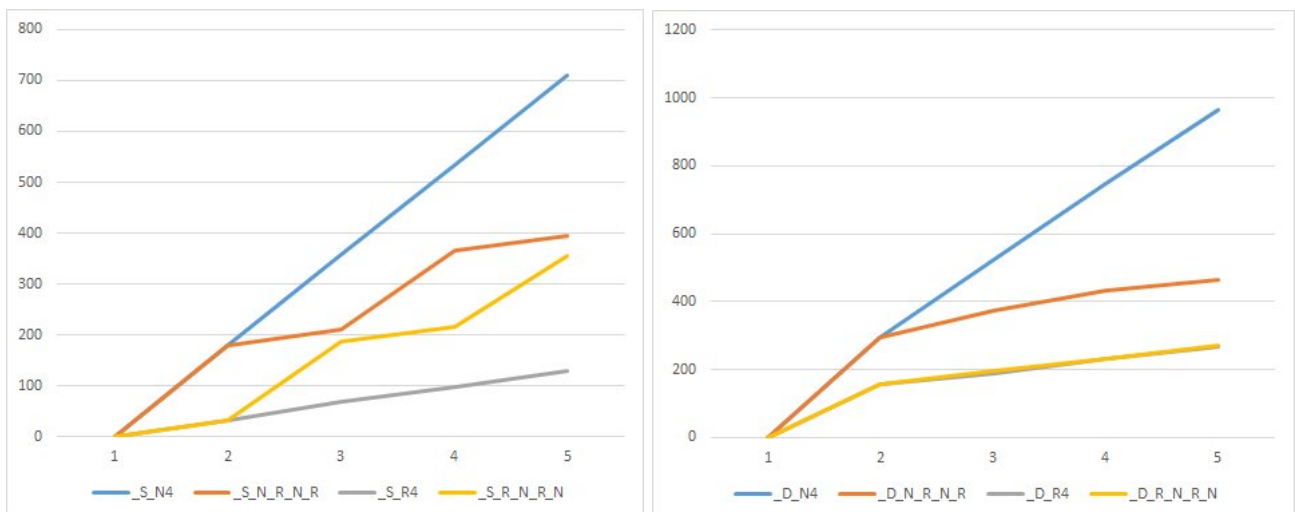


Abbildung 5.7: Kumulierter Zeitaufwand pro Iteration mit statischer bzw. dynamischer Anfangslösung

In den gegenwärtig implementierten Versionen der beiden Verbesserungsstrategien zeigt sich ein immenser Anstieg der Rechenzeiten in Abhängigkeit von der Gartengröße, sodass deren Einsatz bereits für mittelgroße Gärten (40×50) als kritisch anzusehen ist, denn hier können die einzelnen Iterationen bereits eine Laufzeit von 20000 Sekunden aufweisen. Beispielsweise benötigt die Verfahrensreihenfolge *S_R_N_R_N* insgesamt 46500 Sekunden (ca. 13 Stunden).¹ Umso mehr wächst damit die Bedeutung der Eröffnungsverfahren (hierzu siehe auch Kapitel 6).

Die obigen Trendergebnisse müssen noch dahingehend relativiert werden, dass es sich bei der Testreihe um willkürlich ausgewählte Größenordnungen und größtenteils zufällig erzeugte Gartenmodelle handelt. Da zudem die Parametereinstellungen einheitlich vorgenommen wurden und somit auch noch keine konkreten Aussagen über Abhängigkeiten zwischen Lösungsgüte und möglichen Parametervariationen vorliegen, wird keine Allgemeingültigkeit für die obigen Ergebnisse erhoben.

6 Zusammenfassung und Ausblick

Nach den ausgeführten Anwendungen der Verfahren an einer Reihe von Testbeispielen kann festgestellt werden, dass die Laubverteilung in den Gartenmatrizen Auswirkung auf die Ergebnisse der Eröffnungsverfahren hat. Ist die Laubverteilung eher natürlicher Art, also die Menge des Laubs nimmt mit der Entfernung zu den Bäumen ab, dann liefert das statische Eröffnungsverfahren bessere Ergebnisse als das dynamische. Dieses Ergebnis ist auf die Harkstrategie des statischen Verfahrens zurückzuführen, die das Laub stufenartig von außen zum größten Laubhaufen, also Richtung des Baums harkt. Damit werden kleinere Laubmengen öfter geharkt und große Laubmengen weniger oft oder gar nicht, was dazu führt, dass der Harkaufwand gering gehalten wird. Dieses Verfahren ist jedoch bei Gartenmatrizen, die eine zufällige Laubverteilung aufweisen, nicht dienlich. Die stufenartige Harkstrategie bietet hier keine Vorteile, da nicht auf Laubmengen geachtet wird. Das führt dazu, dass auch große Laubmengen über mehrere Felder geharkt werden und zum Teil großer Harkaufwand entsteht. Bei zufälligen Laubverteilungen ist die dynamische Harkstrategie die intelligentere Lösung. Das Verfahren sucht die Harkwege, die den geringsten Harkaufwand gestatten. Die Faustregel ist: Ein Feld wird nur zur Harkfolge hinzugenommen, wenn seine Laubmenge abnimmt oder gleichbleibt. Große Laubmengen werden damit mit diesem Grundsatz seltener geharkt, als geringe Laubmengen.

Die exemplarisch vorgestellten Gartenmatrizen spiegeln relativ extreme Ausprägungen wider. Realistische Gärten wären wohl eine Mischung aus beiden Beispielen. Es wäre also nicht im Vorhinein klar, welches Eröffnungsverfahren angewendet werden soll, um das geringere Harkaufwandsergebnis zu erhalten. Da sich zusätzlich gezeigt hat, dass die Verbesserungsverfahren bereits für mittelgroße Gärten ein kritisches Rechenzeitverhalten aufweisen, rückt die Entwicklung von spezifischen Eröff-

¹Die Tests wurden auf einem PC mit Intel i7-2600, 8GB Ram unter Windows 7-96bit mit MatLab 2015a durchgeführt.

nungsstrategien in den Fokus der weiteren Untersuchungen.

Das Laubharkproblem im Ganzen beinhaltet viele verschiedene Aspekte, die in dieser Ausarbeitung nicht behandelt werden. Die Einbeziehung des Transportaufwands ist sehr einfach modelliert und hat in den Verfahren keinen Einfluss auf den Harkvorgang. Auch der Transport über Ausschlussfelder wurde nicht ausgeschlossen. Der Transport kann bei entsprechender Parametereinstellung einen nicht unwesentlichen Teil des Gesamtaufwandes ausmachen, sodass die Bedeutung von Harkstrategien durchaus in den Hintergrund und die Entwicklung effizienter Transportstrategien in den Vordergrund treten könnten.

Da Gärten selten perfekt rechteckig geformt sind, wäre eine weitere Betrachtung solcher „Sonderformen“ sicher interessant. Die Benutzung von Ausschlussfeldern zur Modellierung dieser Gartenformen erfordert gewiss eine Anpassung der Clusterbildungsverfahren. Eine wesentliche Erweiterung wäre die automatische Einstellung von Harkparametern in der Laufzeit, wie beispielsweise die maximale Laubharkmenge \bar{M} . Die Parameter sind zwar zum Teil auch von rein natürlichen Zuständen abhängig, wie die Stabilität eines Hub-Haufens, können aber die Harklösungen eventuell zum schlechteren beeinflussen. Somit wäre die Idee, einen oder mehrere Parameter durch ein erweitertes Verfahren bestimmen zu lassen, sodass der Gesamtaufwand durch eine „optimierte“ Parametereinstellung minimiert wird.

Dieser Beitrag zeigt eine erste Möglichkeit an, ein Laubharkproblem aufwandsorientiert zu lösen. Die unterschiedlichen Verfahren bieten Kombinationsmöglichkeiten, um den Harkaufwand in diversen Gärten zu verringern.

Literaturverzeichnis

- [Cam96] CAMPBELL, J.F.: Hub location and P-hub median problem. Operations Research, 44:923–935, 1996.
- [HP09] HEKMATFAR, M. und PISHAVAAE, M.: Hub location problem. In: FARAHANI, R.Z. und HEKMATFAR, M. (Herausgeber): Facility location: Concepts, models, algorithms and case studies. 2009.
- [Kli91] KLINCEWICZ, J.G.: Heuristics for the P-hub location problem. European Journal of Operational Research, 53:25–37, 1991.
- [Kru16] KRUSE, H.-J.: Mathematische Modellierung des Laubharkproblems als spezielle Ausprägung von allgemeinen Entsorgungsprozessen. In: KRUSE, H.-J. und LASK, T. (Herausgeber): Angewandte mathematische Modellierung und Optimierung –

Ausgewählte Modelle, Methoden, Fallstudien, Band 4 der Reihe Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Seiten 62–88, 2016. ISSN 2196-6192.

- [Sav92] SAVELSBERGH, M.W.P.: The vehicle routing problem with time windows: Minimizing route duration. *Journal on Computing*, 4:146–154, 1992.
- [Spe15] SPENST, N.: Entwicklung und Implementierung von effizienten Lösungsverfahren für ein spezielles Entsorgungsproblem. Masterarbeit im Studiengang Optimierung und Simulation der Fachhochschule Bielefeld, 2015.
- [VB01] VAN BREEDAM, A.: Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers and Operations Research*, 28:289–315, 2001.
- [WZ98] WIESE, H. und ZELEWSKI, S.: Müllentsorgung und Müllvermeidung: Theoretische und heuristische Ansätze. *Zeitschrift für Betriebswirtschaft*, (ZfB 1/98):145–176, 1998.

Biomimetic Methods for Power Plant Optimization

Friedrich Biegler-König¹

¹Fachbereich Ingenieurwissenschaften und Mathematik, FH Bielefeld

Contact:

friedrich.biegler-koenig@fh-bielefeld.de

Abstract. The energy production optimization of a fleet of power plants involves the solution of a mixed integer problem. Usually this is done by linearization of the continuous and non-linear parts of the problem and subsequent application of a Simplex-type algorithm. Alternatively, heuristic “biomimetic” optimization methods can be used which can also handle nonlinearity. As an example we are proposing an approach to first model power plant blocks with fast Neural Networks and then optimize the operation of multi-block power plants over a period of time using Simulated Annealing and Artificial Bee Colony Algorithm.

Key Words. Evolutionary Methods, Neural Networks, Mixed Integer Optimization, Power Plant Optimization.

1 Introduction

The increase in fossil fuel prices and the introduction of CO₂ credits to European markets have intensified the attempts of providing energy efficient operation modes in existing power plants. Mathematical modeling is an essential tool to provide a basis on which optimization can take place. Optimization then deals with the unit commitment in power generation planning. It consists of scheduling start-up/shut-down decisions and fixing operation levels for a fleet of power generation units. The desired aim is to minimize variable costs and, consequently, to maximize revenues.

This optimization has to take a number of temporal constraints into account:

- time dependency of power prices at the power exchange
- limited availability of some power plant blocks
- costs of start-up and shut-down procedures
- limits for overall emission of CO₂ within a given period of time.

From this it is obvious that a one-point-in-time optimization is not sufficient to tackle this problem. The optimization should rather consider a time period with sufficient time resolution. Because decision variables (power plant on or off) are involved, and every unit has a non-vanishing lower limit of power which it is able to generate, the above-defined problem is a mixed integer optimization problem.

The well understood approach to this type of problem is the use of methods of mixed integer linear programming [Sch03]. It is widely used and allows for optimization with complex temporal constraints and reasonable CPU time consumption. However, recent studies [BD05], [DBN05], [No10] have shown that a nonlinear approach can easily take the current state of the power plant blocks into account, and thus realize an additional potential for optimization. It overcomes the usually extensive computer time consumption of closed nonlinear analytical models by meta-modeling the power plant blocks using artificial neural networks. However, this approach was restricted to a one-point-in-time optimization only. This paper presents an extended approach (compared to the one presented in [BD05]), which is discrete in time and covers 24 hours of energy production with complex constraints. The resulting large mixed integer problem is treated by employing heuristic and biomimetic optimization methods, in this case Simulated Annealing and Artificial Bee Colony algorithm.

2 Approaches to Mixed Integer Optimization

The common approach to optimization in unit commitment uses a linear model of power generation and applies the simplex method of linear programming together with a Branch-and-Bound

approach to solve the integer linear problem. The Simplex method is an iterative method to solve linear problems with constraints. Within a finite number of steps it reaches the solution or proves infeasibility of the problem. The algorithm cannot be applied if some of the variables are restricted to integers. In that case methods are available which apply to a more or less general class of problems. One of these methods is Branch-and-Bound, which consists of three phases. In the branching phase the feasible region is partitioned, in the bounding phase linear simplex is used to find upper and lower bounds of the optimal objective values and in the coordination phase rules are applied for eliminating parts of the feasible region from further consideration.

The above approach uses well-known mathematics but has a serious drawback: it has to eliminate all non-linearities from the optimization, and thus loses essential details. Example: the influence of external parameters such as the cooling water temperature or of internal parameters such as heat surface fouling cannot be taken into account properly.

An alternative approach is the application of so-called “Heuristic Methods” or “Biomimetic Methods”, i.e. methods which emulate the optimization behavior of physical or biological systems. Well-known algorithms are e.g. Genetic Algorithms, Simulated Annealing (SA), Ant Colony Algorithms (see e.g. [GKK04]), or, recently, Artificial Bee Colony (ABC) algorithms [PGK⁺06], [Wik]. All above mentioned algorithms are iterative. They are based on heuristics and bear the advantage that they can easily handle mixed integer problems containing continuous as well as discrete variables. They usually need a high number of iterations to produce good optimal states of the underlying mathematical model. It is however impossible to prove that the computed results are optimal. The following chapter will give an introduction to the simulation approach for power plant blocks, before SA and ABC algorithms are applied to a specific problem in sections 4 and 5 .

3 Fast Simulation of Power Plants

We are considering a fleet of power plants containing four blocks of identical construction (a simple case). Each block can generate power in the range of 108 MW to 360 MW. A block cannot produce less than 108 MW. After a block is switched off, it can only be started up again at considerable costs.

A detailed model of a power plant block was built using Epsilon, a program developed by Sofbid (<http://www.sofbid.com/epsilon/>, see [BP03] and [BP04]) in Zwingenberg, Germany. Epsilon is a simulator specialized in power generating facilities. Epsilon models can be very complex (see figure 1 for a section of a model of a power station block). After constructing the model Epsilon requires about 1 to 10 seconds on a normal desktop computer to simulate a given situation for one block. This is by far not fast enough, since the considered algorithms require many thousands of model evaluations.

Power Plant Optimization

A specific situation of the power plant block is determined by only a few input parameters. The most important parameter is the amount of energy which the block is to produce per time period. The other parameters describe the external conditions and settings by the operational staff:

As a main result, Ebsilon supplies the amount of fuel (in our case coal) required. This in turn determines the main part of the total costs.

As described in [BD05], it is possible to replace the simulator Ebsilon by a much faster meta-model based on Neural Networks. The model-error of fitting a Neural Network to the Ebsilon model has the same magnitude as the model-error of an Ebsilon simulation. The main advantage of the Neural Network model is its response time. It is about 10,000 times faster than Ebsilon (response time: less than 0.0001 seconds). However, even with this speed-up the time used by any algorithm is proportional to the number of model evaluations.

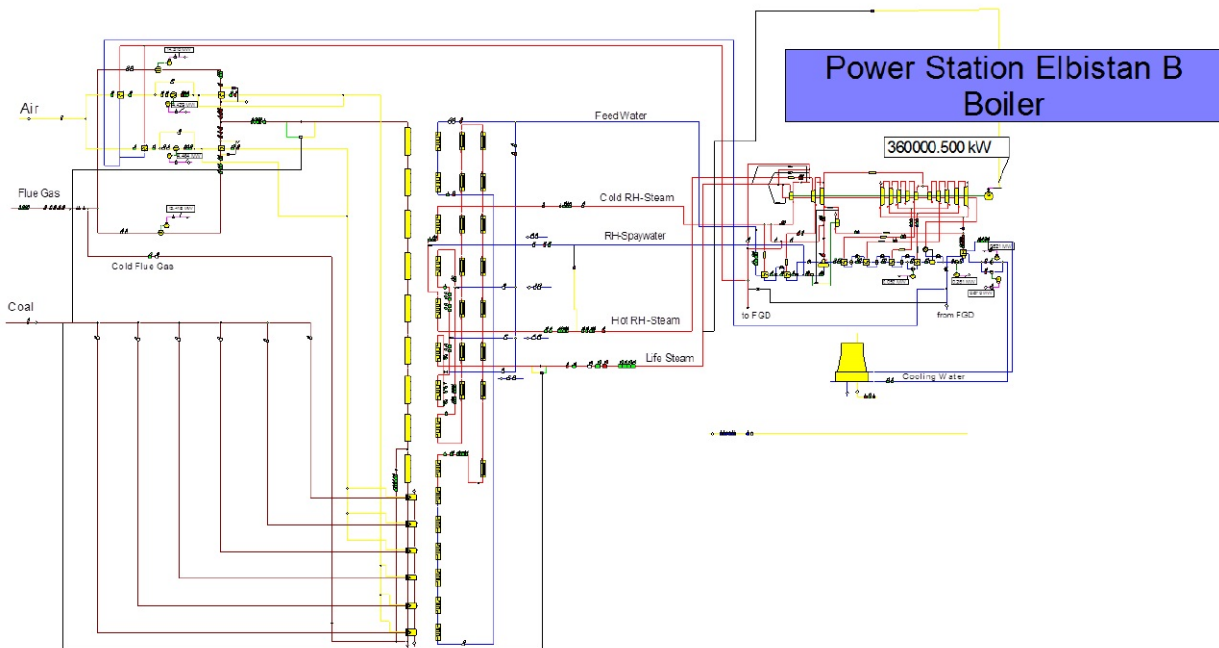


Figure 3.1: Section of Ebsilon Power Station Model

4 Simulated Annealing

Simulated Annealing is based on the imitation of the cooling process of metals.

Here is a brief mathematical description of the algorithm:

Let $F(x_1, \dots, x_N)$ be a target function in N variables. These variables can come from different sources: real numbers, binaries, integers, with or without lower and upper bounds. Additionally, these variables obey a set of restrictions. We assume that F possesses a global minimum whose position we want to determine.

Simulated Annealing is defined by the following iteration:

- For each component x_i of the vector $x = (x_{11}, \dots, x_N)$ define a set of elementary steps C_i whose members change x_i .
- We start with an initial temperature of T_0 , and define a final temperature T_E and a cooling factor $\alpha < 1$. We also have a start vector $x^0 = (x_1^0, \dots, x_N^0)$, which satisfies all constraints. Let, for the beginning, $x^{min} = x^0$ and $F^{min} = F(x^0)$.
- In iteration no. i the following operations will be executed ($i = 1, \dots$):
 1. Until a new candidate vector x_i is found which satisfies all constraints, choose a random component k of vector x^{i-1} and change it using a randomly chosen elementary step from C_i .
 2. Calculate $F(x^i)$.
 3. If $F(x^i) \leq F(x^{i-1})$, set $F^{min} = F(x^i)$. F^{min} is the new optimal value (so far).
 4. If $F(x^i) > F(x^{i-1})$, we accept x^i as the vector for the next iteration with the probability $p(T_i) = \exp((F(x^{i-1}) - F(x^i))/T_i)$. As the algorithm sometimes accepts a worsening of the values, it is able to leave local minima.
 5. If x_i was not accepted, set $x^i = x^{i-1}$.
 6. Set $T_i = \alpha T_{i-1}$ and terminate the algorithm if $T_i < T_E$. The system has “cooled down”.

The iteration process takes longer with larger T_0 and larger α . The probability of finding the correct global minimum increases with the number of performed iteration steps.

Simulated Annealing is not population-based and easy to implement. There are many variations of this standard version of Simulated Annealing (see e.g. [NGH⁺01]). Possible changes are:

- A different probability function $p(T)$ can be chosen.
- Instead of a constant α a cooling function $\alpha(T)$ can be used.

- Elementary steps can be defined as being temperature dependent

Drawbacks of Simulated Annealing:

- The algorithm searches a smaller area than population-based methods
- If cooling too fast, the iteration may end up in a bad local optimum
- Even with slow cooling the iteration may leave a good optimum and end up in a bad one. This can be avoided by memorizing the so far best solution.

5 Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm or Bees algorithm is a recent invention ([PGK⁺06]). It emulates the behavior of a swarm of bees looking for food. A bee hive sends out a certain amount of “scouts” which look for promising food sources (e.g. flower patches). They return to the hive and communicate their findings by performing the “waggle dance”. The scouts then go back to their food sources followed by other bees (more followers for better sources). Less favorable sources are abandoned; the corresponding scouts look elsewhere for more promising places. The observation of the behavior of a bee swarm leads to a population-based swarm intelligence algorithm which can be used to optimize mathematical target functions as defined above.

In pseudo code the basic version for of the ABC algorithm looks like this ([PGK⁺06], [Wik]):

1. Initialize the population with random solutions.
2. Evaluate target function for members of the population.
3. While stopping criteria are not met, form a new population:
 4. Select best sites for neighborhood search (“Scouts”).
 5. Abandon other sites and send their bees to the neighborhood of selected sites.
 6. Determine their target function value.
 7. For each neighborhood select the bee with best target function value.
 8. Distribute all remaining bees randomly over the search space and evaluate their target function value.
9. End While.

Step 8 will ensure that eventually all regions of the search space are considered. Population size and percentage of scouts are the control parameters of the algorithm.

As in Simulated Annealing, elementary steps, i.e. a defined neighborhood of a candidate solution, must be defined.

6 Power Plant Optimization: A Case Study

In the following, SA (see also [Nol07]) and ABC algorithms will be employed to optimize the operation of the described fleet of four power plant blocks for the period of 24 hours.

Every hour a different amount of energy must be produced and these 24 values are provided a day ahead. In reality, the accuracy of these estimates is very high. They comprise information e.g. about time of year and weather forecast.

For each time $t_i, i = 1, \dots, 24$ the production of the described power plant with 4 blocks is described by the vector $x(t_i) = (E_1, E_2, E_3, E_4, b_1, b_2, b_3, b_4)^T$.

b_1, b_2, b_3 and b_4 are binary values and denote whether or not a block is switched off. E_1, E_2, E_3, E_4 are the production rates of the blocks. In every hour of the day they must add up to the given total production. We also have to consider the restrictions of minimal and maximal production rates: $108\text{MW} \leq E_i \leq 360\text{MW}$.

The target function we want to minimize is the cost of energy production accumulated over the day. Since, for every hour in a day, we have a vector of eight variables, the total number of variables in our problem is $24 \times 8 = 192$. In our simple model the total costs consist of the costs for coal and the start-up costs for blocks going on-line. Thus the aim is not only to minimize the production costs in every single hour, but also to keep the number of switch-on-processes of power plant blocks low.

In order to apply the described algorithms we still need to define the elementary steps. Basically, two kinds of steps can be identified:

1. Change of production rate E_k for block no. k without altering the configuration of the power plant. These steps try to optimize a given configuration. This corresponds to solving the continuous part of the problem. In the beginning a maximal stepsize S_{max} is defined. S_{max} is multiplied by a random number between -1 and 1 . This yields the current stepsize. The block number k is also chosen from the set of running blocks.

Afterwards, the restrictions are checked (minimal and maximal production rates, total energy production). If it is not possible to satisfy all boundary conditions, another elementary step must be chosen. This type of elementary step is applied with a probability of 90%.

2. Change of power plant configuration. In this case the binary valued variables are changed. Let n be the number of running blocks. Possible changes of configuration are: Starting up an additional block ($4-n$ alternatives), shutting down a running block (n alternatives), in-

terchanging a running and a pausing block (m alternatives with $m = 3$ if $n = 1$ or $n = 3$, $m = 4$ if $n = 2$, otherwise $m = 0$).

This type of elementary step is applied with a probability of 10%. Among these, the steps are chosen according to the number of alternatives.

After interchanging two blocks, it is not necessary to adjust the production rates, since in our case all blocks have the same limitations. Starting up or shutting down a block makes it necessary to readjust the production rates in order to satisfy the restrictions. If this is not possible, another elementary step must be chosen.

In addition to the elementary steps the following control parameter settings have been used for SA:

- $T_0 = 10,000$.
- $T_E = 10^{-6}$.
- $\alpha = 0.99998$.
- Maximal number of iterations: $it_{max} = 1,000,000$.

The control parameters of ABC algorithm are:

- Population size = 50.
- Percentage of scouts: 10%.
- Number of iterations: $it_{max} = 1,500$.

With this choice of parameters both algorithms found acceptable solutions for the total costs over a day, although, of course, it cannot be verified that the solutions found represent global optima.

Example:

For our example we have used the described 4-block power plant of Elbistan (Turkey), a coal price of 30 Euros for a ton and a start-up price of 30,000 euros for each block. For each hour of the day a required energy production rate is given.

Additionally, the values for heatsurface fouling and cooling temperature for block 1 were changed for the worse; the corresponding values of block 3 were changed for the better. The parameters of blocks 2 and 4 are in between, the values of block 2 are slightly better than those of block 4.

Starting configuration and initial population have been chosen randomly, but satisfy all constraints. Random starting configurations are usually quite bad, since they contain more start-up

processes than necessary (see table 6.1). The first column of the tables contains the time of day, the second the required energy production in kW. Columns 3 to 6 show how the blocks share the production of this energy amount. Tables 6.2 and 6.3 contain the results for SA and ABC algorithms, respectively, with the above settings of parameters

Table 6.1: Initial configuration

Uhrzeit	Vorgabe	Block 1	Block 2	Block 3	Block 4
00:00	530000	360000	170000	0	0
01:00	310000	0	0	0	310000
02:00	200000	200000	0	0	0
03:00	180000	0	0	0	180000
04:00	250000	250000	0	0	0
05:00	320000	0	0	0	320000
06:00	490000	360000	130000	0	0
07:00	650000	0	0	290000	360000
08:00	830000	360000	360000	110000	0
09:00	1030000	0	310000	360000	360000
10:00	1230000	360000	360000	360000	150000
11:00	1130000	282500	282500	282500	282500
12:00	1000000	360000	360000	280000	0
13:00	900000	0	180000	360000	360000
14:00	880000	360000	360000	160000	0
15:00	930000	0	210000	360000	360000
16:00	950000	360000	360000	230000	0
17:00	1000000	0	280000	360000	360000
18:00	1100000	275000	275000	275000	275000
19:00	1240000	160000	360000	360000	360000
20:00	1320000	360000	360000	360000	240000
21:00	1362300	282300	360000	360000	360000
22:00	1223000	360000	360000	360000	143000
23:00	830000	0	110000	360000	360000

Table 6.2: Solution computed with SA

Uhrzeit	Vorgabe	Block 1	Block 2	Block 3	Block 4
00:00	530000	0	108565	211437	209998
01:00	310000	0	0	153338	156662
02:00	200000	0	0	199999	0
03:00	180000	0	0	180000	0
04:00	250000	0	126432	123566	0
05:00	320000	0	145880	174120	0
06:00	490000	0	241964	248036	0
07:00	650000	0	265798	272984	111218
08:00	830000	0	115082	359410	355507
09:00	1030000	0	359500	359540	310959
10:00	1230000	153591	359237	357749	359474
11:00	1130000	108104	339363	341302	341231
12:00	1000000	300353	110047	289906	299694
13:00	900000	110631	266104	267436	255828
14:00	880000	267615	111725	241880	258780
15:00	930000	116764	265973	273476	273786
16:00	950000	275284	129582	276409	268724
17:00	1000000	302995	293152	294674	109179
18:00	1100000	108162	329713	330481	331645
19:00	1240000	161862	359837	358664	359637
20:00	1320000	359020	241419	359825	359734
21:00	1362300	357927	359971	359908	284493
22:00	1223000	144060	359202	360000	360000
23:00	830000	0	357988	359923	112090

Table 6.3: Solution computed with ABC algorithm

Uhrzeit	Vorgabe	Block 1	Block 2	Block 3	Block 4
00:00	530000	0	170000	360000	0
01:00	310000	0	0	309999	0
02:00	200000	0	0	199999	0
03:00	180000	0	0	179998	0
04:00	250000	0	0	250000	0
05:00	320000	0	0	320000	0
06:00	490000	0	129999	360000	0
07:00	650000	0	290000	360000	0
08:00	830000	0	342045	359948	128007
09:00	1030000	0	310001	360000	359999
10:00	1230000	150002	359997	360000	360000
11:00	1130000	146206	263796	359999	360000
12:00	1000000	112163	284973	313019	289845
13:00	900000	154280	235359	351367	158995
14:00	880000	137467	222878	281802	237853
15:00	930000	237643	220202	323749	148406
16:00	950000	146816	251191	360000	191991
17:00	1000000	360000	153117	359999	126884
18:00	1100000	157037	273416	360000	309547
19:00	1240000	160001	360000	360000	359999
20:00	1320000	360000	360000	360000	239999
21:00	1362300	360000	282301	360000	359999
22:00	1223000	143001	360000	360000	359999
23:00	830000	0	338076	360000	131924

The ABC algorithm required about 10 Million evaluations of the underlying neural network, while SA took about half as many evaluations and, subsequently, about half the computation time.

Increasing the number of iterations in SA does not improve the results. The time needed for both methods on a standard PC stays well below 30 minutes, thus is always faster than reality.

Both algorithms supply qualitatively similar solutions and are quite robust to changes in control parameters. They operate block 3, which has the best efficiency, at full capacity. Block 1 is only used if necessary since its efficiency is worst.

SA started with a configuration which costs 1,818,778 euros (configuration of Table 6.1) for the day and lowered it to 1,297,099 euros (configuration of table 6.2). The initial 20 start-up processes were reduced to 3.

ABC algorithm started with a population whose best member costs 1,438,616 euros a day and contains 7 start-up processes. The result configuration of Table 6.3 costs 1,281,726 euros and also contains 3 start-up processes.

The main weakness of SA is that it may leave good solutions and eventually concentrate on less favourable ones. ABC always keeps the so far best solutions and tries to improve on them. This can clearly be seen by comparing the two tables.

7 Conclusion

In this paper we have discussed an approach to power plant optimization which comprises a number of techniques from artificial intelligence as well as from Biomimetics. These techniques were combined with a number of numeric state-of-the-art processes to assemble a concept which is specially customized for power-engineering requirements. In a former paper ([BD05]) we showed that Neural Networks can be trained with high accuracy to reproduce the results of computationally intensive thermodynamic simulation programs on the basis of physical fundamentals. These Neural Networks are meta-models whose response times lie within the range of milliseconds.

On this basis we can use heuristic optimization methods which require a large number of iterations, i.e. a large number of model evaluations. We demonstrated this approach first with Simulated Annealing, a heuristic optimization algorithm, which appears to be suitable for solving the mixed-integer optimization problems arising in power generation planning and which allows an online solution for the non-linear optimization problems.

As second example, we used an Artificial Bee Colony algorithm. This is a population-based heuristic optimization algorithm. In our case it yields better results than Simulated Annealing, even though it needs more computation time.

In an earlier study this concept was successfully applied to one-point-in-time optimization, which already resulted in considerable savings of costs and fuel. The present study shows that a suitable extension of the target function allows for optimization of a model which is discrete in time, covers a whole day of energy production and considers constraints by linking different time intervals. It is clearly shown that the described approach enables nonlinear mixed integer optimization of unit commitment in a fleet of power plants. The methods are easily extended to more complex boundary conditions such as temperature changes in steam piping or the introduction of maintenance intervals for the power station blocks.

Bibliography

- [BD05] Biegler-König, F. and Deeskow, P. Fast simulation and optimization with neural networks. In Proc. 19th European Conference on Modelling and Simulation, Riga, 2005.
- [BP03] Brinkmann, K. and Pawellek, R. Epsilon - examples for the easier design and better operation of power plants. In Proc. Conf. Energy Forum 2003, Sv. Konstantin Varna, 2003.
- [BP04] Brinkmann, K. and Pawellek, R. Optimierte prozessführung von kraftwerksblöcken mit online-werkzeugen: Betriebserfahrungen. In Proc. VDI Tagung "Wissensbasiertes Betriebsmanagement senkt Kosten", Frimmersdorf, 2004.
- [DBN05] Deeskow, P., Biegler-König, F., and Nolle, L. Schnelle optimierung von kraftwerksparkes mit neuronalen netzen. In Proc. 6. VDI-Fachtagung Optimierung in der Energiewirtschaft, Stuttgart, 2005.
- [GKK04] Gerdes, I., Klawonn, F., and Kruse, R. Evolutionäre Algorithmen. Vieweg Verlag, Wiesbadent, 2004.
- [NGH⁺01] Nolle, L, Goodyear, A., Hopgood, A., Picton, P., and Braithwaite, N. On step width adaptation in simulated annealing for continuous parameter optimisation. In Proc. 7th Fuzzy Days, pages 589–598, Dortmund, 2001.
- [Nol07] L. Nolle. Non-linear total energy optimisation of a fleet of power plants. Applications and Innovations in Intelligent Systems XIV, 2007.
- [PGK⁺06] Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., and Zaidi, M. The bees algorithm – a novel tool for complex optimisation problems. In Proceedings of IPROMS 2006 Conference, page 454–461, 2006.

[Sch03] R. Schultz. Course material: Integer programming applied to power systems' generation and operation planning. Technical report, Institut für Mathematik, Gerhard-Mercator-Universität Duisburg, 2003. Preprint 557-2003.

[Wik] Wikipedia. http://en.wikipedia.org/wiki/Bees_algorithm. Accessed: 2015-07-28.

Applications of Shape and Topology Optimization

Svetozara I. Petrova ¹

¹Fachbereich Ingenieurwissenschaften und Mathematik, FH Bielefeld

Contact:

svetozara.petrova@fh-bielefeld.de

Abstract. The problem of structural optimization arises in many industrial and technical applications (e.g. computer aided design, aeronautics, computational fluid dynamics, high power electronics, etc.). A typical problem of structural optimization is to minimize a function (called objective) over a set of geometrical or behavioral restrictions (called constraints). The set of structural variables includes the so-called state and design parameters, and the problem consists in computing optimal values of the design parameters, such that they minimize the specific objective function. Sizing, shape, and topology optimization problems are different types in structural optimization. In the optimization algorithm, we are typically faced with nonconvex nonlinear optimization problems with both equality and inequality constraints on the state variables and design parameters. We focus on the numerical solution of the resulting nonlinear programming problem by using primal-dual interior-point algorithms, Newton's method and line-search approach. This study emphasizes on optimization problems with partial differential equations as constraints. Numerical results for design and topology optimization with applications to materials science and engineering are presented and discussed.

Keywords. structural optimization, primal-dual method, inexact Newton-like solver, asymptotic homogenization, eddy currents, finite elements

1 Introduction

Structural optimization has recently become of increasing interest in computer aided design and optimization of composite structures in materials science and specific devices in electrical engineering (cf., e.g., [Ben95] and the references therein). A typical problem of structural optimization is to minimize a function (called *objective*, *cost* or *criterion* function) over a set of geometrical or behavioral requirements (called *constraints*). The set of structural parameters includes the so-called *state* and *design* parameters, and the problem consists in computing optimal values of the design parameters, such that they minimize the specific objective function. For the choice of the objective, various criteria can be taken into account: mechanical properties (minimum compliance), loading (bending, tension, torsion), thermal properties (shock resistance), technological properties (minimum weight), economical properties (cheapness of production), etc.

Sizing, shape, and topology optimization problems are different types in structural optimization. Detailed classification of these problems is given, for instance, in [OT83]. In the sizing problems, the goal is to find the optimal thickness distribution of a given material structure. The main difficulty in shape optimization problems arises from the fact that the geometry of a structure is a design variable which means, in particular, that the discretization model associated with the structure has to be changed in the process of optimization, see [Roz88]. In the topology optimization of solid structures we are interested in the determination of the optimal placement of material in space, i.e., one has to determine which points of space are material and which points should remain void (no material). Hence, the main goal of these problems is to find the location of holes and the connectivity of the domain, see [BS03].

In the optimization algorithm, we are typically faced with constrained nonconvex nonlinear minimization problems with both equality and inequality constraints on the state variables and design parameters. Details for the solution methods, based on the primal–dual Newton–like algorithms, are given in Section 2. We consider two practical applications of optimal structural design, described in Section 3 and Section 4. Numerous numerical experiments are presented and discussed.

2 Primal–dual Newton interior–point method

For the discretized optimization problem we use the primal–dual Newton interior–point methods, recently a topic of intensive research [BHN99, ETTZ96, FGW02, GOW98, HP04, VS99]. The main idea of these methods is to generate iteratively approximations of the solution which strictly satisfy the inequality constraints.

2.1 General nonlinear optimization problem

We consider the following general constrained nonlinear nonconvex programming problem with both equality and inequality constraints

$$\min_{\mathbf{x} \in \mathcal{R}^n} f(\mathbf{x}), \quad (2.1)$$

subject to

$$\mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \quad (2.2)$$

where $f : \mathcal{R}^n \rightarrow \mathcal{R}$, $\mathbf{h} : \mathcal{R}^n \rightarrow \mathcal{R}^m$, $m < n$, and $\mathbf{g} : \mathcal{R}^n \rightarrow \mathcal{R}^l$ are assumed to be twice Lipschitz continuously differentiable. Note that the constraints (2.2) have to be understood componentwise.

The *Lagrangian function* associated with (2.1)-(2.2) is defined by

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + \mathbf{y}^T \mathbf{h}(\mathbf{x}) - \mathbf{z}^T \mathbf{g}(\mathbf{x}), \quad (2.3)$$

where $\mathbf{y} \in \mathcal{R}^m$ and $\mathbf{z} \in \mathcal{R}^l$ are the Lagrange multipliers for the equality and inequality constraints, respectively.

The first-order Karush-Kuhn-Tucker (KKT) necessary conditions for optimality of (2.1)-(2.2) read

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0}, \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \quad Z \mathbf{g}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0},$$

where

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \nabla f(\mathbf{x}) + \sum_{i=1}^m y_i \nabla h_i(\mathbf{x}) - \sum_{i=1}^l z_i \nabla g_i(\mathbf{x}) \quad (2.4)$$

is the gradient of the Lagrangian function and Z is the diagonal matrix with a diagonal \mathbf{z} . We also consider the Hessian of the Lagrangian with respect to \mathbf{x} defined by

$$\nabla_{\mathbf{x}}^2 \mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \nabla^2 f(\mathbf{x}) + \sum_{i=1}^m y_i \nabla^2 h_i(\mathbf{x}) - \sum_{i=1}^l z_i \nabla^2 g_i(\mathbf{x}), \quad (2.5)$$

where $\nabla^2 f(\mathbf{x})$, $\nabla^2 h_i(\mathbf{x})$, $1 \leq i \leq m$, $\nabla^2 g_i(\mathbf{x})$, $1 \leq i \leq l$ stand for the Hessians of $f(\mathbf{x})$, $h_i(\mathbf{x})$, and $g_i(\mathbf{x})$, respectively. Denote by

$$\mathcal{A}(\mathbf{x}) = \{i \mid g_i(\mathbf{x}) = 0, i = 1, \dots, l\}$$

the set of all indices for which the inequality constraints are equal to zero at \mathbf{x} . We are interested in finding local minimizers of our optimization problem (2.1)-(2.2). Assume that at least one such point \mathbf{x}^* exists satisfying the conditions:

- **Feasibility:** $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ and $\mathbf{g}(\mathbf{x}^*) \geq \mathbf{0}$.

- **Regularity:** The set $\{\nabla h_1(\mathbf{x}^*), \dots, \nabla h_m(\mathbf{x}^*)\} \cup \{\nabla g_i(\mathbf{x}^*), i \in \mathcal{A}(\mathbf{x}^*)\}$ of gradients of equality and active inequality constraints is linearly independent.
- **Smoothness:** The Hessian matrices $\nabla^2 f(\mathbf{x})$, $\nabla^2 h_i(\mathbf{x})$, $1 \leq i \leq m$, and $\nabla^2 g_i(\mathbf{x})$, $1 \leq i \leq l$, exist and are locally Lipschitz continuous at \mathbf{x}^* .
- **Second-order sufficiency condition:** $\boldsymbol{\eta}^T \nabla_{\mathbf{x}}^2 \mathcal{L}(\mathbf{x}^*) \boldsymbol{\eta} > 0$ for all vectors $\boldsymbol{\eta} \neq 0$ satisfying $\nabla h_i(\mathbf{x}^*)^T \boldsymbol{\eta} = 0$, $1 \leq i \leq m$, and $\nabla g_i(\mathbf{x}^*)^T \boldsymbol{\eta} = 0$, $i \in \mathcal{A}(\mathbf{x}^*)$.
- **Strict complementarity:** $z_i^* > 0$ if $g_i(\mathbf{x}^*) = 0$, $1 \leq i \leq l$.

Well-known approaches from the optimization theory for handling problems with inequality constraints are, for instance, the slack variable approach, the active set strategy, and the logarithmic barrier function approach. Each of these approaches results in a nonlinear programming problem with only equality constraints. For example, in the first approach, the constraint $\mathbf{g}(\mathbf{x}) \geq 0$ can be replaced by $\mathbf{g}(\mathbf{x}) - \mathbf{s} = 0$, $\mathbf{s} \geq 0$ by adding a nonnegative slack variable to each of the inequality constraints. Transformation of the original inequality problem into an equality one, by adding slacks, have been a frequently applied tool in scientific computations in the past twenty years and recently used in cf., [BHN99, ETTZ96, VS99]. The introduction of slack variables is associated with a small amount of additional work and storage, since they do not enter the objective function and are constrained by simple bounds. The second, active set strategy, approach in nonlinear programming is directly related to the idea of the simplex method in linear programming. At each iterative step from this approach, applying, for example, Newton's method, one has to define which constraints are active at the solution and treat them as equality constraints by ignoring the others. The third approach was used in our practical implementations and we explain it in detail in the next subsection.

2.2 Logarithmic barrier interior–point method

The logarithmic barrier function method was first introduced in [Fis55] and later on popularized by [FM69] in the late sixties of the last century. The basic idea of this method is to replace the optimization problem (2.1)-(2.2) with the following equality constrained optimization problem

$$\min_{\mathbf{x} \in \mathcal{R}^n} \beta^{(\rho)}(\mathbf{x}), \quad (2.6)$$

subject to

$$\mathbf{h}(\mathbf{x}) = 0, \quad (2.7)$$

where ρ is a positive scalar, called *barrier parameter*, and

$$\beta^{(\rho)}(\mathbf{x}) = f(\mathbf{x}) - \rho \sum_{i=1}^l \log g_i(\mathbf{x}) \quad (2.8)$$

is often referred to as a *barrier function*. To insure existence of the logarithmic terms in (2.8) we implicitly require $g_i(\mathbf{x}) > 0$, $1 \leq i \leq l$. In such a way, we get a family of subproblems depending on ρ for which it is well-known that under the assumptions conditions from Subsection 2.1 the solution of (2.6)-(2.7) converges to a solution of the original problem (2.1)-(2.2) as ρ decreases to zero (cf., [FM69]). This method obviously is an interior-point method since it keeps the sequence of iterating solutions strictly feasible with respect to the inequality constraints. Note that the logarithmic terms serve as a barrier and result in finding a solution $\mathbf{x}^{(\rho)}$ such that $g(\mathbf{x}^{(\rho)}) > 0$. The solution points $\mathbf{x}^{(\rho)}$ parameterized by ρ define the so-called *central path* or also called *barrier trajectory*.

The gradient of (2.8) is given by

$$\nabla \beta^{(\rho)}(\mathbf{x}) = \nabla f(\mathbf{x}) - \sum_{i=1}^l \frac{\rho}{g_i(\mathbf{x})} \nabla g_i(\mathbf{x})$$

and the Hessian of $\beta^{(\rho)}(x)$ is defined by

$$\nabla^2 \beta^{(\rho)}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) - \sum_{i=1}^l \frac{\rho}{g_i(\mathbf{x})} \nabla^2 g_i(\mathbf{x}) + \sum_{i=1}^l \frac{\rho}{g_i^2(\mathbf{x})} \nabla g_i(\mathbf{x})(\nabla g_i(\mathbf{x}))^T. \quad (2.9)$$

The Lagrangian function associated with (2.6)-(2.7) is

$$\mathcal{L}^{(\rho)}(\mathbf{x}, \mathbf{y}) = \beta^{(\rho)}(\mathbf{x}) + \mathbf{y}^T \mathbf{h}(\mathbf{x}) = f(\mathbf{x}) - \rho \sum_{i=1}^l \log g_i(\mathbf{x}) + \mathbf{y}^T \mathbf{h}(\mathbf{x})$$

and the gradient of $\mathcal{L}^{(\rho)}(\mathbf{x}, \mathbf{y})$ with respect to \mathbf{x} is given by

$$\nabla_{\mathbf{x}} \mathcal{L}^{(\rho)}(\mathbf{x}, \mathbf{y}) = \nabla f(\mathbf{x}) - \sum_{i=1}^l \frac{\rho}{g_i(\mathbf{x})} \nabla g_i(\mathbf{x}) + \sum_{i=1}^m y_i \nabla h_i(\mathbf{x}). \quad (2.10)$$

The logarithmic barrier function method consists now of generating a sequence of iterative solutions $\{\mathbf{x}\} = \{\mathbf{x}^{(\rho)}\}$, local minimizers of the equality constrained subproblems, with $\rho > 0$ decreasing at each iteration. Taking into account the first-order optimality conditions and especially $\nabla_{\mathbf{x}} \mathcal{L}^{(\rho)}(\mathbf{x}, \mathbf{y}) = 0$, we see that convergence of $\{\mathbf{x}^{(\rho)}\}$ to an optimal solution \mathbf{x}^* requires that

$$\lim_{\rho \rightarrow 0} y_i^{(\rho)} = y_i^*, \quad 1 \leq i \leq m \quad \text{and} \quad \lim_{\rho \rightarrow 0} \frac{\rho}{g_i(\mathbf{x}^{(\rho)})} = z_i^*, \quad 1 \leq i \leq l, \quad (2.11)$$

where $\{y_i^*\}$ and $\{z_i^*\}$ are the Lagrange multipliers associated with the equality and inequality constraints $g_i(\mathbf{x}^{(\rho)}) > 0$, respectively. From $g_i(\mathbf{x}^{(\rho)}) \rightarrow 0$ and the second relation in (2.11) we get $\rho/g_i^2(\mathbf{x}^{(\rho)}) \rightarrow \infty$ and hence, the Hessian of the logarithmic barrier function (2.9) would become arbitrarily large. Comparing now relations (2.4) and (2.10), we see that $\rho/g_i(\mathbf{x}^{(\rho)})$ serves as a Lagrange multiplier for the inequality constraints. Thus, we can introduce an auxiliary variable

$z_i = z_i^{(\rho)} = \rho/g_i(\mathbf{x}^{(\rho)})$, $1 \leq i \leq l$ which can also be written in the form $z_i^{(\rho)} g_i(\mathbf{x}^{(\rho)}) = \rho$. The last relation is usually called *perturbed complementarity* and can be used as a remedy, so that the differentiation will not create ill-conditioning.

We formulate now the perturbed KKT conditions for the logarithmic barrier function problem (2.6)-(2.7), namely

$$\nabla f(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})\mathbf{y} - \nabla \mathbf{g}(\mathbf{x})\mathbf{z} = \mathbf{0}, \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad Z\mathbf{g}(\mathbf{x}) = \rho\bar{\mathbf{e}}, \quad \mathbf{g}(\mathbf{x}) > \mathbf{0}. \quad (2.12)$$

In matrix-vector notations, (2.12) results in the following nonlinear equation with $n + m + l$ components

$$\mathbf{F}^{(\rho)}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0} \quad \text{with} \quad \mathbf{F}^{(\rho)}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \begin{pmatrix} \mathbf{t} + J_{\text{eq}}^T \mathbf{y} - J_{\text{in}}^T \mathbf{z} \\ \mathbf{h} \\ G\mathbf{z} - \rho\bar{\mathbf{e}} \end{pmatrix}, \quad (2.13)$$

where $\mathbf{F}^{(\rho)} = \nabla \mathcal{L}^{(\rho)}$ is the gradient of the Lagrangian function with respect to \mathbf{x}, \mathbf{y} , and \mathbf{z} ; $\mathbf{t} = \nabla f(\mathbf{x})$ is the gradient of the objective function, J_{eq} is the Jacobian $m \times n$ matrix of the equality constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ and J_{in} is the Jacobian $l \times n$ matrix of the inequality constraints $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$. In the last equation of (2.13) we have denoted $G = \text{diag}(g_i)$, $g_i > 0$, $1 \leq i \leq l$ and $\bar{\mathbf{e}} = (1, 1, \dots, 1)^T$. Note that at each iteration we have three sets of unknowns: the primal variable \mathbf{x} , the dual variable \mathbf{y} , and the perturbed complementarity variable \mathbf{z} which we consider independently.

Denote the unknown solution by $\Phi = (\mathbf{x}, \mathbf{y}, \mathbf{z})^T$. We apply the Newton method to the nonlinear system (2.13) and find the increments $\Delta\Phi = (\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{z})^T$ by means of

$$K\Delta\Phi = -\mathbf{F}^{(\rho)}(\Phi), \quad (2.14)$$

which is often referred to as a *primal-dual system*. The vector $\Delta\Phi$ is called *search direction*. The so-called *primal-dual matrix* $K = \left(\mathbf{F}^{(\rho)}\right)'(\Phi)$ of second derivatives of the Lagrangian function is defined as follows

$$K = \begin{pmatrix} H & J_{\text{eq}}^T & -J_{\text{in}}^T \\ J_{\text{eq}} & 0 & 0 \\ ZJ_{\text{in}} & 0 & G \end{pmatrix}, \quad (2.15)$$

where the Hessian of the Lagrangian function $H = \nabla_x^2 \mathcal{L}$ is given by (2.5). Note that the matrix K is sparse, nonsymmetric, independent of ρ , and usually well-conditioned in a sense that its condition number is limited when $\rho \rightarrow 0$ (see [Wri98] for more details). In the case of convex optimization (i.e., convex objective function $f(\mathbf{x})$, linear equality constraints $\mathbf{h}(\mathbf{x})$, and concave inequality constraints $\mathbf{g}(\mathbf{x})$), the Hessian matrix H is positive semidefinite. The properties of the

Hessian matrix for inequality constrained optimization problem with logarithmic barrier function method are discussed in [FGW02].

One possible way for solving (2.14) is to symmetrize K taking into account the fact that Z and G are diagonal matrices. This method is proposed in [FGS96] and results in the following symmetric matrix

$$\hat{K} = \begin{pmatrix} H & J_{\text{eq}}^T & -J_{\text{in}}^T \\ J_{\text{eq}} & 0 & 0 \\ -J_{\text{in}} & 0 & -Z^{-1}G \end{pmatrix},$$

which is strongly ill-conditioned with some diagonal elements becoming unbounded as $\rho \rightarrow 0$. In particular, for the active inequality constraints, the diagonal entries of $Z^{-1}G$ go to zero, and for the inactive constraints they go to infinity. As the iterates converge, the ill-conditioning of K increases, but it was shown in [FGS96] that the primal-dual solution of the optimization problem is actually independent of the size of the large diagonal elements and can be found by using, for instance, a symmetric indefinite factorization of the primal-dual system.

Another alternative way for solving (2.14) which we use in our practical applications is to eliminate the (1,3) block of (2.15), i.e., due to $\mathbf{g}(\mathbf{x}) > 0$, we eliminate $\Delta\mathbf{z}$ from the third equation of (2.14)

$$\Delta\mathbf{z} = -\mathbf{z} + G^{-1}(\rho\bar{\mathbf{e}} - ZJ_{\text{in}}\Delta\mathbf{x}) \quad (2.16)$$

and replace it in the first equation. This method produces a symmetric linear system with $n + m$ equations of the form

$$\begin{pmatrix} \tilde{H} & J_{\text{eq}}^T \\ J_{\text{eq}} & 0 \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{pmatrix} = - \begin{pmatrix} \mathbf{t} + J_{\text{eq}}^T\mathbf{y} - \rho J_{\text{in}}^T G^{-1}\bar{\mathbf{e}} \\ \mathbf{h} \end{pmatrix}, \quad (2.17)$$

where $\tilde{H} = H + J_{\text{in}}^T G^{-1} Z J_{\text{in}}$ is often referred to as a *condensed* primal-dual Hessian. Thus, (2.17) is called a *condensed* primal-dual system. A detailed analysis of the properties of the condensed primal-dual matrix can be found in [Wri98] where it was shown that the inherent ill-conditioning of the reduced primal-dual matrix is usually benign and does not influence the accuracy of the solution.

Various methods for solving (2.17) and finding the primal-dual steps $(\Delta\mathbf{x}, \Delta\mathbf{y})$ are proposed in the literature (cg., e.g., [ETT96, GOW98, Wri98]). Note that one needs a reliable and efficient solver of (2.17), since the condensed primal-dual system is solved at every iteration of the optimization loop. In practice, we apply transforming iterations (see [Wit89]) to find the increments.

After finding the solution of (2.17), the algorithm proceeds iteratively from an initial point $(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \mathbf{z}^{(0)})$ through a sequence of points determined from the search directions described by (2.16) and (2.17) as follows

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_x^{(k)} \Delta \mathbf{x}, \quad \mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha_y^{(k)} \Delta \mathbf{y}, \quad \mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \alpha_z^{(k)} \Delta \mathbf{z}.$$

The parameters $\alpha_x^{(k)}, \alpha_y^{(k)}, \alpha_z^{(k)} \in (0, 1]$ are called *steplengths* and their choice at each iteration is a critical feature of the algorithm to find a local minimizer of the optimization problem.

2.3 Merit functions. Computing the steplengths

In all optimization algorithms it is important to have a reasonable way of deciding whether the new iterate is better than the previous one, i.e., it is essential to measure appropriately the progress in finding a local solution. Merit functions of different types have been a subject of great interest over the past years (see, e.g., [ETTZ96, GOW98, Wri98]). The main idea of a merit function is to ensure simultaneously a progress toward a local minimizer and toward feasibility. The method of choosing $\alpha^{(k)}$ at each iteration becomes more complicated in general nonlinear programming problems as it is well-known that the Newton method may diverge when the initial estimate of the solution is bad.

Two versions of the Newton method can be applied, namely, the *trust-region* and the *line-search* approach. The first method has recently been applied in, e.g., [BHN99]. Typical for this method is to find a step $\mathbf{d}^{(k)}$ which is restricted to a set, called the *trust region*. This set is practically obtained by limiting $|\mathbf{d}^{(k)}| \leq r^{(k)}$, where $r^{(k)}$ is the trust region radius. At each iteration, $r^{(k)}$ is updated according to how successful the step has been. For instance, if the a priori chosen merit function M decreases, we accept the step $\mathbf{d}^{(k)}$, update the solution $\Phi^{(k+1)} = \Phi^{(k)} + \mathbf{d}^{(k)}$ and possibly increase the trust region radius $r^{(k)}$. Otherwise, we decrease $r^{(k)}$ by a damping factor, e.g., $r^{(k)} = r^{(k)}/2$ and compute again the step $\mathbf{d}^{(k)}$.

We apply the second variant of the Newton method, the *line-search* approach. Once the solution $\Delta \Phi^{(k)}$ of (2.14) has been determined, we find a steplength $\alpha^{(k)} > 0$ such that $\Phi^{(k+1)} = \Phi^{(k)} + \alpha^{(k)} \Delta \Phi^{(k)}$ measuring a progress in minimization at each iteration and reducing the merit function in the sense $M(\Phi^{(k+1)}) < M(\Phi^{(k)})$. The ideal value $\alpha^{(k)} = 1$ may not always happen so that various modifications of the basic Newton method have to be implemented. The following basic model algorithm can be considered:

- S1. If the conditions for convergence are satisfied, the algorithm terminates with $\Phi^{(k)}$ as the solution;
- S2. Compute a search direction $\Delta \Phi^{(k)}$ solving (2.14);
- S3. Compute the steplength $\alpha^{(k)} > 0$ for which $M(\Phi^{(k)} + \alpha^{(k)} \Delta \Phi^{(k)}) < M(\Phi^{(k)})$;

- S4. Update the estimate for the minimum by $\Phi^{(k+1)} := \Phi^{(k)} + \alpha^{(k)} \Delta \Phi^{(k)}$, $k := k + 1$, and go back to step S1.

A standard convergence monitor in nonlinear programming is to choose the Euclidean norm $\|\mathbf{F}^{(\rho)}(\mathbf{x}, \mathbf{y}, \mathbf{z})\|$ of the residual produced by the KKT conditions (2.13) as a merit function. However, in many practical implementations, this choice of the merit function is not sufficient, since it does not allow to tell the difference between a local minimizer and a stationary nonminimizing point. The KKT conditions are necessary optimality conditions and hence, the optimization problem (2.1)-(2.2) and the nonlinear problem (2.13) are not equivalent, i.e., the Newton method may find solutions of (2.13) which do not minimize the objective function $f(\mathbf{x})$. Therefore, in order to find simultaneously solutions of both problems, a better approach is to rely on a hierarchy of two merit functions. In general, the choice of merit functions in nonlinear constrained optimization problems is complicated. Several ideas have recently been proposed in the context of primal–dual interior methods (cf., e.g., [BHN99, ETTZ96, FGW02]). In particular, our *primary merit function* is related to those suggested in [GOW98] and is chosen as a modified augmented Lagrangian incorporating the logarithmic barrier function (2.8) as follows

$$M := M(\mathbf{x}, \mathbf{y}, \rho, \rho_A) = f(\mathbf{x}) - \rho \sum_{i=1}^l \log g_i(x) + \mathbf{y}^T \mathbf{h}(\mathbf{x}) + \frac{1}{2} \rho_A \mathbf{h}(\mathbf{x})^T \mathbf{h}(\mathbf{x}), \quad (2.18)$$

where ρ_A is a positive parameter. Our purpose now is to satisfy the descent conditions and to guarantee a reduction of the merit function in the sense that each iterate should be an improved estimate of the solution of (2.6)-(2.7). Note that a descent is sought only with respect to \mathbf{x} taking into account the original optimization problem. A standard way to achieve $M(\mathbf{x} + \alpha \Delta \mathbf{x}, \mathbf{y}, \rho, \rho_A) < M(\mathbf{x}, \mathbf{y}, \rho, \rho_A)$ is to require that $\Delta \mathbf{x}$ is a descent direction, i.e., $\Delta \mathbf{x}^T \nabla_{\mathbf{x}} M < 0$, where $\nabla_{\mathbf{x}} M$ is the gradient of the primary merit function with respect to the primal variable \mathbf{x} . In particular, we have

$$\begin{aligned} \Delta \mathbf{x}^T \nabla_{\mathbf{x}} M &= \Delta \mathbf{x}^T (\mathbf{t} - \rho J_{\text{in}}^T G^{-1} \bar{\mathbf{e}} + J_{\text{eq}}^T \mathbf{y} + \rho_A J_{\text{eq}}^T \mathbf{h}) \\ &= \Delta \mathbf{x}^T (\mathbf{t} - \rho J_{\text{in}}^T G^{-1} \bar{\mathbf{e}}) - \mathbf{h}^T \mathbf{y} - \rho_A \mathbf{h}^T \mathbf{h}, \end{aligned} \quad (2.19)$$

due to $J_{\text{eq}} \Delta \mathbf{x} = -\mathbf{h}$ from the second equation of (2.17). Hence, $\Delta \mathbf{x}^T \nabla_{\mathbf{x}} M < 0$ can be satisfied if the augmented Lagrangian parameter ρ_A is sufficiently large, namely

$$\rho_A > \frac{\Delta \mathbf{x}^T (\mathbf{t} - \rho J_{\text{in}}^T G^{-1} \bar{\mathbf{e}}) - \mathbf{h}^T \mathbf{y}}{\mathbf{h}^T \mathbf{h}}.$$

Hence, ρ_A can be changed within the optimization loop, if $\Delta \mathbf{x}$ is not a descent direction. In our algorithm, we choose

$$\rho_A = \min \left(\frac{5}{\mathbf{h}^T \mathbf{h}} (\Delta \mathbf{x}^T (\mathbf{t} - \rho J_{\text{in}}^T G^{-1} \bar{\mathbf{e}}) - \mathbf{h}^T \mathbf{y}), 100 \right) \quad (2.20)$$

in the case $\Delta \mathbf{x}^T \nabla_{\mathbf{x}} M \geq 0$ and continue the loop (see [GOW98, HP04] for details).

For the *secondary merit function* we choose the l_2 - norm of the residual with respect to the perturbed KKT-conditions (2.13). We apply the Newton method and choose the steplengths to strictly satisfy the inequality constraints $\mathbf{g}(\mathbf{x}) > 0$ and the complementarity constraints $\mathbf{z} > 0$. Hence, the first requirement for the line-search approach is to insure a strict feasibility. Let $\hat{\alpha}$ and $\hat{\gamma}$ be separate steplengths defined as follows

$$\hat{\alpha} = \max\{\alpha | \mathbf{g}(\mathbf{x}) + \alpha J_{\text{in}} \Delta \mathbf{x} \geq 0\}, \quad \hat{\gamma} = \max\{\gamma | \mathbf{z} + \gamma \Delta \mathbf{z} \geq 0\}.$$

Since we maintain interior (i.e., strict feasible) iterates, usually we take a parameter $\tau \in (0, 1)$ bounded strongly away from unity and define $\alpha = \min(1, \tau \hat{\alpha})$ and $\gamma = \min(1, \tau \hat{\gamma})$. We use the same steplength γ for the Lagrange multiplier \mathbf{y} . In practice, both merit functions are used by means of the following strategy: If the steplengths α and γ lead to a reduction of M , they are accepted. If M does not decrease, we check the secondary merit function. If the latter decreases, the steplengths are accepted; otherwise damp the Newton steps by a certain factor and continue the procedure. The barrier parameter $\rho > 0$ is updated by decreasing values until an approximate solution of the nonlinear problem is obtained (cf., e.g., [ETTZ96, GOW98, HP04]). We rely on a *watchdog strategy* (see [CLPP82]) to ensure progress in finding a local minimizer. If after some fixed number of iterations there is no reduction of M , the augmented Lagrangian parameter ρ_A is chosen sufficiently large in accordance with (2.20).

3 Microstructural ceramics

A recent idea in biomimetical applications is to take advantage of naturally grown wood in the production of high performance ceramics. Typical feature of the wood structure is the system of the tracheidal cells which provide the transportation path for water and minerals within the living tree. This open porous system is accessible for infiltration of various metals. Among the large variety of ceramic composites, biomorphic cellular silicon carbide (SiC) ceramics from wood have been recently produced and investigated, see [OTH⁺95]. They can be used as filters and catalysts in chemical processing, heat insulation structures, thermally and mechanically loaded lightweight structures, and medical implants (for instance, for bone substitution). The conversion of naturally grown wood to highly porous SiC ceramics is done by a process called *biotemplating* which includes two processing steps: high-temperature pyrolysis and silicon (Si) infiltration. More details for the producing scheme can be found in [GLK98].

The geometry of the final ceramics, i.e., the widths and lengths of the different (carbon- and silicon carbide-) layers, can be determined very precisely by an appropriate tuning of the process parameters. The question is how to choose these microstructural geometrical data in order to

achieve an optimal performance with respect to a merit criterion. From a mathematical point of view, this issue represents a shape optimization problem where the state variables are subject to the underlying elasticity equations and the microstructural data (widths/lengths of the material layers) serve as design variables. We rely on the homogenization approach, assuming a periodically distributed microstructure, and further apply the optimization to the homogenized model.

3.1 The homogenized model

In this section, we briefly explain the derivation of the homogenized computational model on the macroscale by using the asymptotic homogenization theory, cf. [JKO94]. Homogenization has been successfully used in the last three decades for solving multi-scale problems on computational regions occupied by heterogeneous microstructural materials.

The homogenized model for our original heterogeneous material occupying the domain Ω , $\Omega \subset \mathcal{R}^d$, $d = 2, 3$, is illustrated in Figure 3.1. The main idea of the homogenization is to replace the heterogeneous material by an equivalent homogenized material, extracting the information for the material properties of the various microstructural constituents (or different phases).

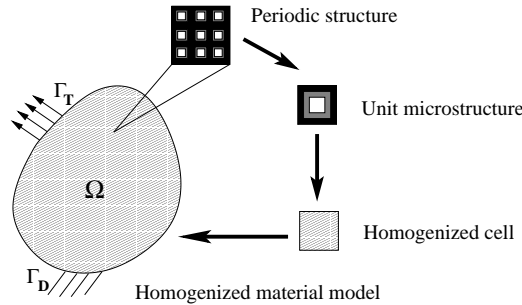


Figure 3.1: The macroscopic homogenized material model

The microscopic and macroscopic models are considered simultaneously supposing a strong scale separation, i.e., a large gap in length scale between the macroscopic component and the microstructure. A main assumption in the homogenization approach is that the original heterogeneous material workpiece is composed of periodically distributed microstructures of various constituents. To couple properly the micro- and macro-scales, we choose a representative volume element (RVE) or a unit microstructure $Y = [0, 1]^d$, $d = 2, 3$, consisting of an outer layer of carbon (C), interior layer of silicon carbide (SiC), and a centered pore channel (no material).

The homogenization method requires to find periodic functions ξ^{kl} satisfying the following problem in a weak formulation to be solved in the microscopic unit cell Y

$$\int_Y E_{ijpq}(\mathbf{y}) \frac{\partial \xi_p^{kl}}{\partial y_q} \frac{\partial \phi_i}{\partial y_j} dY = \int_Y E_{ijkl}(\mathbf{y}) \frac{\partial \phi_i}{\partial y_j} dY, \quad (3.1)$$

where $\phi \in \mathbf{H}^1(Y)$ is an arbitrary Y -periodic variational function. Here, we have adopted the Einstein summation rule on repeated indices. The function $\boldsymbol{\xi}^{kl}$, also referred to as the characteristic displacement, is found by solving (3.1) in Y with periodic boundary conditions. After computing $\boldsymbol{\xi}^{kl}$, one defines the homogenized coefficients by the following formulas (we refer to [JKO94] for details)

$$E_{ijkl}^H = \frac{1}{|Y|} \int_Y \left(E_{ijkl}(\mathbf{y}) - E_{ijpq}(\mathbf{y}) \frac{\partial \xi_p^{kl}}{\partial y_q} \right) dY. \quad (3.2)$$

Due to the symmetry conditions for the elasticity coefficients, the fourth order homogenized elasticity tensor $\mathbf{E}^H = (E_{ijkl}^H)$ can be written as a symmetric and usually dense matrix. The two-dimensional homogenized tensor can be considered as a 3×3 matrix and the three-dimensional homogenized tensor, respectively, as a 6×6 matrix.

3.2 The optimization problem

Our goal is to optimize the mechanical performances of the ceramic composites (such as the compliance or the bending strength) taking into account technological and problem specific constraints on the state and design parameters. The state variables \mathbf{u} are the nodal values of the components of the discrete displacement vector and the design variables $\boldsymbol{\alpha}$ are chosen as the microstructural data determining the geometry of the periodicity cell (widths and lengths of the different materials layers forming the cell walls). The geometrical properties of the final ceramics are not fixed but can be changed and precisely tuned within the processing.

Our shape optimization problem reads: Find $(\mathbf{u}, \boldsymbol{\alpha}) \in \mathcal{R}^N \times \mathcal{R}^M$ such that

$$J(\mathbf{u}, \boldsymbol{\alpha}) = \inf_{\mathbf{v}, \boldsymbol{\beta}} J(\mathbf{v}, \boldsymbol{\beta}), \quad (3.3)$$

subjected to the following equality and inequality constraints

$$A(\boldsymbol{\alpha})\mathbf{u} = \mathbf{f}, \quad g(\boldsymbol{\alpha}) := \sum_{i=1}^M \alpha_i = C, \quad \alpha_{\min} \bar{\mathbf{e}} \leq \boldsymbol{\alpha} \leq \alpha_{\max} \bar{\mathbf{e}}, \quad (3.4)$$

where $J(\mathbf{u}, \boldsymbol{\alpha})$ is the mean compliance of the structure, $A(\boldsymbol{\alpha})$ is the stiffness matrix corresponding to the homogenized equilibrium equation, \mathbf{u} is the discrete homogenized displacement vector, \mathbf{f} is the discrete load vector, and $\bar{\mathbf{e}} = (1, 1, \dots, 1) \in \mathcal{R}^M$. In our particular case, the number of layers M is equal to two. Note that α_{\min} and α_{\max} are technologically motivated lower and upper bounds for the design parameters. In the unit microstructure Y , we take the limits $\alpha_{\min} = 0$, $\alpha_{\max} = 0.5$, and $0 < C \leq 0.5$.

3.3 Numerical experiments

We present now some computational results concerning the microscopic problem to find the homogenized elasticity coefficients and the macroscopic shape optimization problem. We suppose linear elasticity with homogeneous and isotropic constituents in terms of carbon and SiC. The Young modulus E (in GPa) and the Poisson ratio ν of our two materials are, respectively, $E = 10$, $\nu = 0.22$ for carbon and $E = 410$, $\nu = 0.14$ for SiC.

The computation of the characteristic displacement fields ξ^{kl} and the homogenized elasticity coefficients (3.2) requires the solution of linear elastic boundary value problems with the periodicity cell Y as the computational domain. The elasticity equation (3.1) is solved numerically using a conforming finite element discretization of the periodicity cell Y by linear basis functions. Since the periodic displacements $\xi^{kl} = \xi^{lk}$ are symmetric, the equation (3.1) is computed numerically 3 times in the case $d = 2$ and respectively, 6 times in the case $d = 3$.

Denote the global density of the solid material part in the microstructure by μ , $0 < \mu < 1$. Note that the density of the tracheidal cells of the wood essentially depends on the growth of the tree. If μ is relatively small, we speak about an *early wood* (grown in spring and summer) and respectively, about *late wood* (grown in autumn and winter) for values of μ , close to 1.

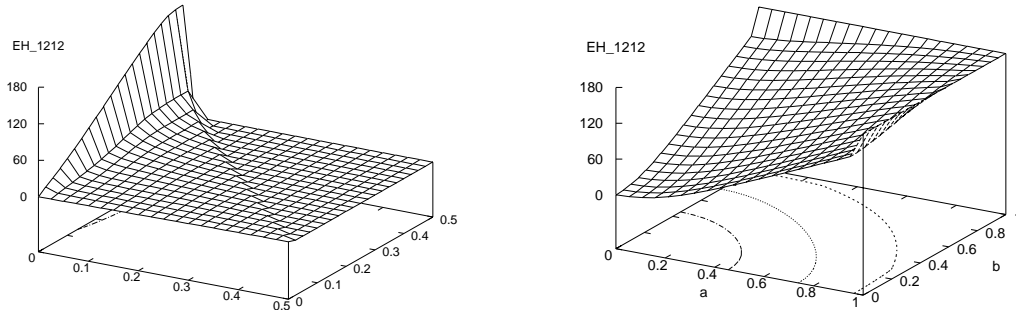


Figure 3.2: Homogenized coefficient E_{1212}^H w.r.t.: a) the widths of carbon and SiC layers (square hole); b) the sizes $1 - a$ and $1 - b$ of the rectangular hole

We present some numerical experiments on a plane microstructure ($d = 2$). Assume that the material layers in the periodicity cell have equal widths from all sides of the cell. Denote by α_i , $i = 1, 2$, the widths of the carbon and SiC layers, respectively. Figure 3.2a) illustrates the behavior of the homogenized coefficient E_{1212}^H in case of square hole versus α_1 and α_2 which vary between 0 and 0.5. We compute the effective coefficients E_{ijkl}^H only for a fixed number of values of the design parameters (e.g., 20×20 grid as shown on Figure 3.2) and then interpolate the values by splines. We demonstrate the behavior, for instance, of E_{1212}^H depending on a rectangular hole $[1 - a] \times [1 - b]$, see Figure 3.2b). Note that $a = b = 0$ represents a complete void, $a = b = 1$ realizes a complete solid

material, and $0 < a < 1$, $0 < b < 1$ characterize a general porous material. We consider in this example the case when the carbon has completely reacted with the SiC which strongly concerns the so-called *pure biomorphic SiC-ceramics*.

Figure 3.3 displays the dependence of the homogenized elasticity coefficients on the density μ of the cell. In particular, we show this behavior versus the width of the SiC layer in case of pure SiC-ceramics. Figure 3.3a) shows the behavior of the effective coefficients for early wood ($0 \leq \alpha_2 \leq 0.15$, $\mu = 51\%$) and Figure 3.3b) demonstrates the coefficients for late wood ($0 \leq \alpha_2 \leq 0.3$, $\mu = 84\%$). One can easily observe from both pictures on this figure a highly nonlinear behavior of the homogenized coefficients.

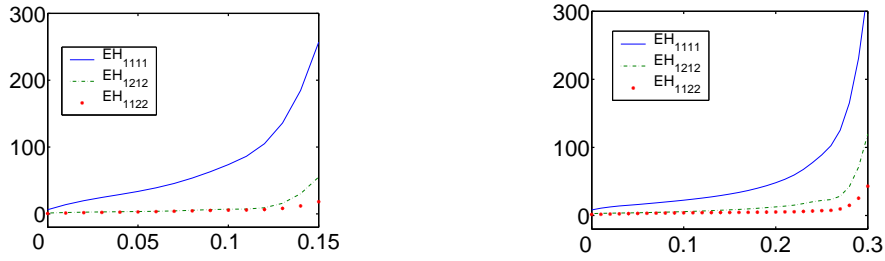


Figure 3.3: Homogenized coefficients w.r.t. the width α_2 for pure SiC-ceramic: a) early wood, density 51% b) late wood, density 84%

Table 3.1: Convergence results for biomorphic microcellular SiC ceramics

$\alpha_1^{(0)}$	$\alpha_2^{(0)}$	C	ITER	α_1	α_2	ρ	M	$\ \mathbf{F}^{(\rho)}\ _2$	$\ \mathbf{v}\ _2$
0.05	0.05	0.3	11	3.6e-12	0.3	1.3e-17	1.24	9.63e-6	e-10
0.1	0.1	0.3	11	5.5e-14	0.3	3.0e-21	1.24	1.03e-6	e-12
0.1	0.1	0.4	12	1.6e-16	0.4	1.2e-26	0.85	8.63e-9	e-14
0.2	0.2	0.1	16	5.5e-17	0.1	2.2e-25	7.73	2.23e-8	e-13
0.2	0.2	0.2	13	1.0e-16	0.2	5.3e-26	2.34	1.54e-8	e-14
0.2	0.2	0.3	11	2.5e-16	0.3	6.7e-26	1.24	1.79e-8	e-14
0.24	0.24	0.15	11	5.4e-15	0.15	4.1e-12	3.81	4.99e-7	e-12
0.3	0.1	0.4	11	1.3e-12	0.4	8.5e-19	0.85	5.07e-6	e-10
0.4	0.05	0.1	17	9.8e-15	0.1	6.9e-21	7.73	9.49e-7	e-11

In Table 3.1 we report some numerical results from running the optimization code varying the constant C with respect to (3.4). Our purpose is to find the optimal lengths of the layers in the composite material and to show the convergence behavior of the optimization algorithm. The domain Ω is chosen to be a circle which corresponds naturally to a cross section of the original wood

structure. We have fixed the discretization and vary the initial values for the lengths of the carbon and SiC layers denoted, respectively, by $\alpha_1^{(0)}$ and $\alpha_2^{(0)}$. We report the number of iterations ITER to get convergence, the optimal lengths α_1 and α_2 of the carbon and SiC layers, the last value of the barrier parameter ρ , the final value of the primary merit function M , the l_2 -norm of the residual, and the l_2 -norm of the complementarity conditions $\mathbf{v} = (\mathbf{z}, \mathbf{w})$ at the last iteration.

We see from the experiments that the optimal length α_1 of the carbon layer in all the runs is very close to zero, i.e., the solid part of the body is entirely occupied by a silicon carbide layer due to the higher stiffness of this material.

4 Optimal design of high–power devices

4.1 The eddy current equations

In this section, we focus on electromagnetic fields in the low frequency regime which can be described by the quasistationary limit of Maxwell’s equations also known as the eddy current equations

$$\frac{\partial \mathbf{B}}{\partial t} + \mathbf{curl} \mathbf{E} = \mathbf{0}, \quad \mathbf{div} \mathbf{B} = 0, \quad \mathbf{curl} \mathbf{H} = \mathbf{J}, \quad (4.1)$$

$$\mathbf{B} = \mu \mathbf{H}, \quad \mathbf{J} = \sigma \mathbf{E}. \quad (4.2)$$

In the system of equations (4.1)-(4.2) we have denoted by \mathbf{E} and \mathbf{H} the electric and the magnetic field and, respectively, by \mathbf{B} and \mathbf{J} the magnetic induction and the current density. The scalar parameters μ and σ refer to the magnetic permeability and the electric conductivity (see, e.g., [LLP84]).

We consider a two-dimensional model with a current density given by

$$\mathbf{J} = (J_1(x_1, x_2, t), J_2(x_1, x_2, t), 0),$$

which suggests, in particular, the following form of the vector fields \mathbf{E} , \mathbf{H} , and \mathbf{B}

$$\mathbf{E} = (E_1(x_1, x_2, t), E_2(x_1, x_2, t), 0), \quad (4.3)$$

$$\mathbf{H} = (0, 0, H(x_1, x_2, t)), \quad (4.4)$$

$$\mathbf{B} = (0, 0, B(x_1, x_2, t)).$$

We consider now the alternative formulation in terms of the electric (also called scalar) potential φ and the magnetic (also called vector) potential \mathbf{A} according to

$$\mathbf{E} = -\mathbf{grad} \varphi - \frac{\partial \mathbf{A}}{\partial t}, \quad \mathbf{B} = \mathbf{curl} \mathbf{A}.$$

This model can be treated in a given spatial domain $\Omega \subset \mathcal{R}^d$, $d = 2, 3$ with a boundary $\partial\Omega$. Then, introducing the additional condition $\operatorname{div} \mathbf{A} = 0$ and taking into account that $\operatorname{div} \mathbf{J} = 0$ in the interior of the domain, (4.1)-(4.2) give rise to the following coupled system of PDEs for the electric potential φ and the magnetic potential \mathbf{A}

$$\operatorname{div}(\sigma \mathbf{grad} \varphi) = 0 \quad \text{in } \Omega, \quad (4.5)$$

$$\sigma \mathbf{n} \cdot \mathbf{grad} \varphi = \begin{cases} I_\nu & \text{on } \Gamma_\nu \subset \partial\Omega \\ 0 & \text{elsewhere} \end{cases}, \quad (4.6)$$

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \mathbf{curl} \mu^{-1} \mathbf{curl} \mathbf{A} = \begin{cases} -\sigma \mathbf{grad} \varphi & \text{in } \Omega \\ 0 & \text{in } \mathbf{R}^3 \setminus \bar{\Omega} \end{cases}. \quad (4.7)$$

In equation (4.6), we refer to I_ν as the fluxes associated with the contacts $\Gamma_\nu \subset \partial\Omega$, $1 \leq \nu \leq N_c$ with a total number of contacts N_c . The fluxes have to satisfy the compatibility condition

$$\sum_{\nu=1}^{N_c} I_\nu = 0. \quad (4.8)$$

Note that the equation (4.7) is considered with appropriate initial and boundary conditions.

The electric energy dissipation given by the Joule-Lenz law reads as follows

$$f(\varphi, \sigma, \mathbf{A}) := \int_{\Omega} \mathbf{J} \cdot \mathbf{E} \, dx. \quad (4.9)$$

In the stationary regime, (4.9) reduces to

$$f(\varphi, \sigma) = - \int_{\Omega} \mathbf{J} \cdot \mathbf{grad} \varphi \, dx = - \int_{\Omega} \operatorname{div}(\varphi \mathbf{J}) \, dx. \quad (4.10)$$

The last equality in (4.10) follows from

$$\operatorname{div}(\varphi \mathbf{J}) = \mathbf{J} \cdot \mathbf{grad} \varphi + \varphi \operatorname{div} \mathbf{J}$$

with $\operatorname{div} \mathbf{J} = 0$. Using the Gauss theorem and the Neumann boundary conditions from (4.6) we get

$$f(\varphi, \sigma) = - \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{J} \varphi \, ds = \sum_{\nu=1}^{N_c} \int_{\Gamma_\nu} I_\nu \varphi \, ds. \quad (4.11)$$

The last expression is considered as an objective functional in our optimization problem. Our purpose is to minimize the electric energy dissipation.

4.2 The optimization problem

Define the equation for the electric potential (4.5) in a weak formulation as follows

$$a(\varphi, v) = L(v), \quad \text{for } v \in U, \quad (4.12)$$

where the bilinear energy form is

$$a(\varphi, v) = \int_{\Omega} \mathbf{grad} v \cdot \sigma \mathbf{grad} \varphi \, dx,$$

$L(v)$ is the linear load form and U is the space of all admissible solutions of (4.5).

Then, the problem to minimize the energy dissipation given by the objective functional (4.11) is to find

$$\inf_{\varphi, \sigma} f(\varphi, \sigma) = \inf_{\varphi, \sigma} \sum_{\nu=1}^{N_e} \int_{\Gamma_{\nu}} I_{\nu} \varphi \, ds, \quad (4.13)$$

subject to the following constraints

$$\begin{aligned} a(\varphi, v) &= L(v) && \text{with BCs from (4.6),} \\ \int_{\Omega} \sigma \, dx &= C && \text{(mass constraint),} \\ \sigma_{\min} &\leq \sigma \leq \sigma_{\max} && \text{(conductivity box constraint).} \end{aligned} \quad (4.14)$$

Here, σ_{\min} and σ_{\max} are a priori given positive limits for the conductivity and C is a fixed given constant. In general formulations of nonlinear programming problems, the objective function f and the inequality constraints are supposed to be twice continuously differentiable. This requirement is obviously satisfied in our case.

Note that we solve the constrained optimization problem with the partial differential equation (4.5) for φ incorporated as a part of the constraints. By means of this formulation, we arrive at the basic idea of the so called one-shot methods which stand in contrast to many standard optimization approaches considering the optimization process and the solution of the differential equation separately. Our experience shows that this simultaneous optimization approach together with a numerical solution of the partial differential equation reduces essentially the overall computational complexity of the resulting optimization algorithm, see [Pet09].

4.3 Level-set approach

We are interested in optimal distribution of the conductivity in a prescribed structural domain $\Omega \subset \mathcal{R}^d$, $d = 2, 3$ whose topology can be changed. To formulate the level-set approach for finding the optimal design, we consider a fixed reference domain $D \subset \mathcal{R}^d$ and suppose that D always

contains the computational design domain Ω which changes within the optimization process, i.e. $\Omega \subseteq D$.

In the level-set framework, one defines a scalar function $\Phi : D \rightarrow \mathcal{R}$ (also referred to as a level-set function) which represents the design boundary $\partial\Omega$ implicitly by

$$\partial\Omega = \{\mathbf{x} | \mathbf{x} \in D, \Phi(\mathbf{x}) = 0\}.$$

Note that $\Phi(\mathbf{x})$ is one-dimensional higher function than the represented boundary $\partial\Omega$. Thus, one can determine each part of the design domain as follows

$$\Phi(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \partial\Omega \cup D, \quad (4.15)$$

$$\Phi(\mathbf{x}) > 0, \quad \forall \mathbf{x} \in \Omega \setminus \partial\Omega, \quad (4.16)$$

$$\Phi(\mathbf{x}) < 0, \quad \forall \mathbf{x} \in D \setminus \Omega. \quad (4.17)$$

The boundary is embedded implicitly as the zero level-set (4.15) of $\Phi(\mathbf{x})$, while the interior of the structure is presented by (4.16) and the exterior by (4.17). Note that the boundary $\partial\Omega$ is totally manipulated through the zero level-set function. Hence, all admissible shapes have the form $\Omega = \{\mathbf{x} | \mathbf{x} \in D, \Phi(\mathbf{x}) \geq 0\}$, see Fig. 4.1.

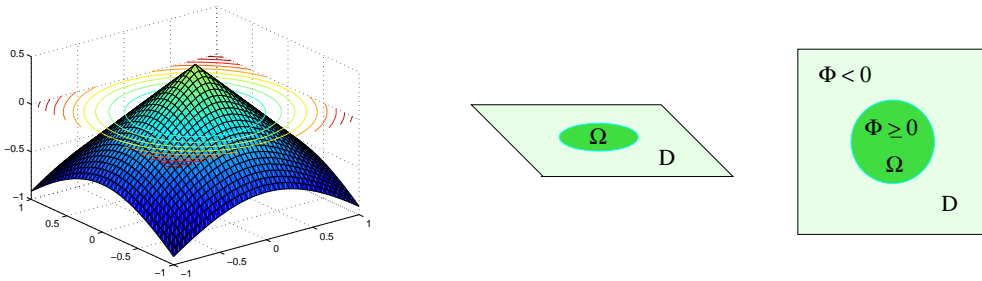


Figure 4.1: a) Level-set representation; b) Zero level-set; c) Design domain $\Omega \subseteq D$

If we suppose that the zero level-set moves in the normal direction to itself, then the level-set function changes dynamically in time, i.e.

$$\Phi(\mathbf{x}(t), t) = k \quad \text{for any } \mathbf{x} \in \Omega(t), \quad (4.18)$$

where $k = 0$ for $\mathbf{x} \in \partial\Omega(t)$. By differentiation of (4.18) with respect to time t and applying the chain rule, one arrives at the so called Hamilton-Jacobi-type equation

$$\frac{\partial\Phi(\mathbf{x}, t)}{\partial t} + \nabla\Phi(\mathbf{x}, t) \frac{d\mathbf{x}}{dt} = 0. \quad (4.19)$$

More details can be found in [OS88]. The latter equation can also be written in the form

$$\frac{\partial \Phi(\mathbf{x}, t)}{\partial t} = -\nabla \Phi(\mathbf{x}, t) \frac{d\mathbf{x}}{dt} \equiv -\mathbf{v} \nabla \Phi(\mathbf{x}, t), \quad (4.20)$$

where $\mathbf{v} := d\mathbf{x}/dt$ denotes the speed of the zero level-set, which is related to the optimized objective functional. Thus, the structural optimization process can be treated as the movement of a point on the boundary which is exactly driven by the objective. For recent implementations of this method in structural optimization, we refer to [AJT04, AA06, Set99, WWG03].

Our optimization problem for this application can be formulated by means of level-set model as follows. Find:

$$\inf_{\Phi} f(\varphi, \sigma, \Phi) = \inf_{\Phi} \sum_{\nu=1}^{N_c} \int_D I_{\nu} \varphi \delta(\Phi) |\nabla \Phi| dx, \quad (4.21)$$

subject to the following constraints

$$\begin{aligned} a(\varphi, v, \Phi) &= L(v, \Phi) && \text{with BCs from (4.6),} \\ \int_D \sigma \bar{H}(\Phi) dx &= C && \text{(mass constraint),} \\ \sigma_{\min} &\leq \sigma \leq \sigma_{\max} && \text{(conductivity box constraint).} \end{aligned} \quad (4.22)$$

The energy form is now defined as

$$a(\varphi, v, \Phi) = \int_D \mathbf{grad} v \cdot \sigma \mathbf{grad} \varphi \bar{H}(\Phi) dx.$$

In this formulation we use minimization of the objective functional with respect to the level-set function Φ , the Heaviside function $\bar{H}(\Phi)$, and the Dirac function $\delta(\Phi)$, cf., e.g., [WWG03].

4.4 Numerical experiments

We present now some computational results concerning our second application. The design domain Ω is chosen as a rectangle occupied by an isotropic conductive medium. The domain is decomposed by uniform quadrilateral finite elements. The rotated bilinear (also referred to Rannacher and Turek) basis functions are used for the discrete model. Comparable finite element formulations of eddy current systems can be found in, cf. [ABN00, BP95, KT93].

A quadrature rule (exact for polynomials of degree three) has been used to compute the global stiffness matrix corresponding to the elliptic differential equation as a first equality constraint. All the computations have been done with preliminary given values for the range of the conductivity. We have chosen $\sigma_{\min} = 0.01$, $\sigma_{\max} = 1$, and a constant C computed according to an initial homogeneous distribution with a conductivity $\sigma = 0.45$. The preconditioned conjugate gradient method (PCG) has been applied to solve systems with the stiffness matrix.

The primal-dual interior-point method with logarithmic barrier functions is implemented in the numerical experiments. Some computational approaches for the choice of the barrier parameter ρ are discussed in [HP04]. The optimization process is computationally unstable when ρ is made too small. We compute the material distribution for various number of contacts N_c . Due to the compatibility condition (4.8), we always have that the current inflow at the lower port(s) is equal to the current outflow at the upper port(s).

The final optimal design is visualized in Fig. 4.2 for $N_c = 2$ and in Fig. 4.3 for $N_c = 3$ and $N_c = 6$. In the plane Oxy one can see the contour of the final design domain. On the axis Oz we observe the computed values of the conductivity running from σ_{\min} up to σ_{\max} . For comparable applications of topology optimization of eddy current systems with different objectives, design parameters, and constraints we refer the reader to [KKW04, SWH07].

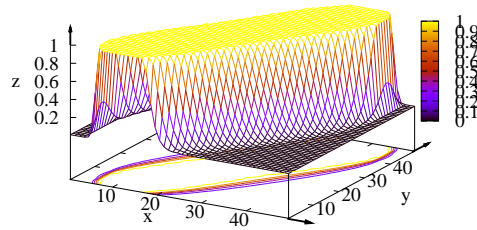


Figure 4.2: Material distribution (50×50 mesh, 2 contacts)

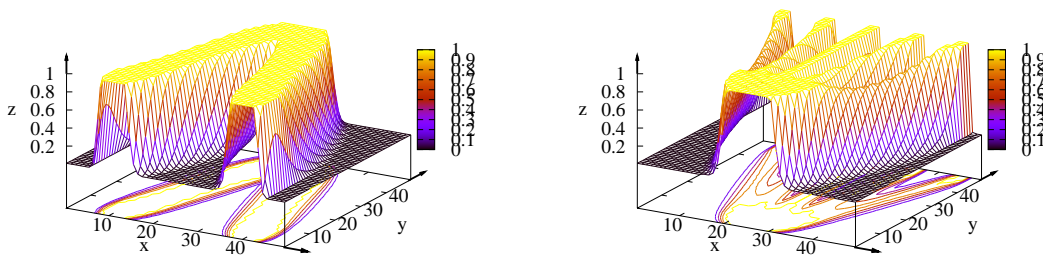


Figure 4.3: Material distribution, 50×50 mesh, a) 3 contacts; b) 6 contacts)

Acknowledgements: The author is thankful to Ronald H.W. Hoppe for the numerous useful discussions and research ideas in the last years. This work has been partially supported by the DAAD PPP Norway (Project ID 57067781) within BMBF funds and the FSP AMMO at the University of Applied Sciences Bielefeld, Germany.

Bibliography

- [AA06] Amstutz, S. and Andrä, H. Structural optimization using sensitivity analysis and a level-set method. J. Comput. Phys., 1216:573–589, 2006.
- [ABN00] Ammari, H., Buffa, A., and Nédélec, J.-C. A justification of eddy currents model for the maxwell equations. SIAM J. Appl. Math., 60:1805–1823, 2000.
- [AJT04] Allaire, G., Jouve, F., and Toader, A.-M. Structural optimization using sensitivity analysis and a level-set method. J. Comput. Phys., 194:363–393, 2004.
- [Ben95] M.P. Bendsøe. Optimization of Structural Topology, Shape, and Material. Springer, Berlin, 1995.
- [BHN99] Byrd, R.H., Hribarr, M.E., and Nocedal, J. An interior point algorithm for large scale nonlinear programming. SIAM J. Optim., 9(4):877–900, 1999.
- [BP95] Biro, O. and Preis, K. Various fem formulations for the calculation of transient 3d eddy currents in nonlinear media. IEEE Transactions on Magnetics, 31:1307–1312, 1995.
- [BS03] Bendsøe, M.P. and Sigmund, O. Topology Optimization, Theory, Methods, and Applications. Springer, New York, 2003.
- [CLPP82] Chamberlain, R.M., Lemaréchal, C., Pedersen, H.C., and Powell, M.J.D. The watchdog technique for forcing convergence in algorithms for constrained optimization. Math. Progr. Study, 16:1–17, 1982.
- [ETTZ96] El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., and Zhang, Y. On the formulation and theory of the Newton interior-point method for nonlinear programming. J. Optim. Theory Appl., 89:507–541, 1996.
- [FGS96] Forsgren, A., Gill, P.E., and Shinnerl, J.R. Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization. SIAM J. Matrix Anal. Appl., 17:187–211, 1996.
- [FGW02] Forsgren, A., Gill, P.E., and Wright, M.H. Interior methods for nonlinear optimization. SIAM Review, 44:525–597, 2002.
- [Fis55] K.R. Fisch. The Logarithmic Potential Method of Convex Programming. University Institute of Economics, Oslo, Norway, 1955. Memorandum.
- [FM69] Fiacco, A.V. and McCormick, G.P. Nonlinear Programming. Sequential Unconstrained Minimization Techniques. John Wiley and Sons, New York, New York, 1969. Republished by SIAM, Philadelphia, Pennsylvania, 1990.

- [GLK98] Greil, P., Liftka, T., and Kaindl, A. Biomorphic cellular silicon carbide ceramics from wood: I. processing and microstructure. Europ. Ceramic Soc., 18:1961–1973, 1998.
- [GOW98] Gay, D.M., Overton, M.L., and Wright, M.H. A primal-dual interior method for nonconvex nonlinear programming. In Y. Yuan, editor, Advances in Nonlinear Programming, pages 31–56. Kluwer, Dordrecht, Holland, 1998.
- [HP04] Hoppe, R.H. and Petrova, S.I. Primal-dual newton interior-point methods in shape and topology optimization. Numer. Linear Algebra Appl., 11:413–429, 2004.
- [JKO94] Jikov, V.V., Kozlov, S.M., and Oleinik, O.A. Homogenization of Differential Operators and Integral Functionals. Springer, 1994.
- [KKW04] Kang, J., Kim, C., and Wang, S. Reliability-based topology optimization for electromagnetic systems. COMPEL, 23(3):715–723, 2004.
- [KT93] Krawczyk, A. and Tegopoulos, J.A. Numerical Modelling of Eddy Currents, volume 32 of Monographs in Electrical and Electronic Engineering. Oxford University Press, 1993.
- [LLP84] Landau, L.D., Lifshitz, E.M., and Pitaevskii, L.P. Electrodynamics of Continuous Media, volume 8. 2 edition, 1984.
- [OS88] Osher, S. and Sethian, J.A. Front propagation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. J. Comp. Phys., 78:12–49, 1988.
- [OT83] Olhoff, N. and Taylor, J.E. On structural optimization. Appl. Mech., 50:1139–1151, 1983.
- [OTH⁺95] Ota, T., Takahashi, M., Hibi, T., Ozawa, M., Suzuki, S., Hikichi, Y., and Suzuki, H. Biomimetic process for producing SiC wood. J. Amer. Ceram. Soc., 78:3409–3411, 1995.
- [Pet09] S.I. Petrova. Applications of one-shot methods in pdes constrained optimization. Math. Comput. Simul., 80(3):581–597, 2009.
- [Roz88] G.I.N. Rozvany. Structural Design via Optimality Criteria. Kluwer, Dordrecht, 1988.
- [Set99] J.A. Sethian. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science. Cambridge University Press, 1999.
- [SWH07] Shim, H., Wang, S., and Hameyer, K. Topology optimization of magnetothermal systems considering eddy current as joule heat. IEEE Trans. Magn., 43(4):1617–1620, 2007.

- [VS99] Vanderbei, R.J. and Shanno, D.F. An interior-point algorithm for nonconvex nonlinear programming. Comput. Optim. Appl., 13:231–252, 1999.
- [Wit89] G. Wittum. On the convergence of multigrid methods with transforming smoothers. theory with applications to the navier-stokes equations. Numer. Math., 57:25–38, 1989.
- [Wri98] M.H. Wright. Ill-conditioning and computational error in interior methods for nonlinear programming. SIAM J. Optim., 9:84–111, 1998.
- [WWG03] Wang, M.Y., Wang, X., and Guo, D. A level-set method for structural topology optimization. Comput. Meth. Appl. Mech. Eng., 192(1-2):227–246, 2003.

Wer und Was ist AMMO?

Im Jahre 2008 gründeten fünf Wissenschaftlerinnen und Wissenschaftler den Forschungs- und Entwicklungsschwerpunkt *Angewandte Mathematische Modellierung und Optimierung* (FSP AMMO). Mittlerweile besteht die Gruppe aus zwölf Personen.



Das AMMO-Team (von links): Prof. Dr. Dr. Rainer Ueckerdt, Dipl.-Math. Willi Braun, Prof. Dr. Svetozara Petrova, Prof. Dr. Claudia Cottin, Timo Lask (M.Sc.), Prof. Dr. Bernhard Bachmann, Prof. Dr. Hermann-Josef Kruse, Dipl.-Ing. Jens Schönbohm, Prof. Dr. Friedrich Biegler-König, Lennart Ochel (M.Sc.), Dr. Sabrina Proß (auf dem Foto fehlt: Vitalij Ruge (M.Sc.))

Wer und Was ist AMMO?

Die AMMO-Mitglieder sind in der folgenden Liste in alphabetischer Reihenfolge aufgeführt:

Prof. Dr. Bernhard Bachmann	Gründungsmitglied
Prof. Dr. Friedrich Biegler-König	Gründungsmitglied, bis 2013 stellvertretender Sprecher
Dipl.-Math. Willi Braun	seit 2012
Prof. Dr. Claudia Cottin	Gründungsmitglied
Prof. Dr. Hermann-Josef Kruse	Gründungsmitglied
Timo Lask (M.Sc.)	seit 2015
Lennart Ochel (M.Sc.)	seit 2013
Prof. Dr. Svetozara Petrova	seit 2009, 2013 - 2015 stellvertretende Sprecherin, seit 9/2015 Sprecherin
Dr. Sabrina Proß	seit 2009, seit 9/2015 stellvertretende Sprecherin
Vitalij Ruge (M.Sc.)	seit 2013
Dipl.-Ing. Jens Schönbohm	seit 2009
Prof. Dr. Dr. Rainer Ueckerdt	Gründungsmitglied, bis 9/2015 Sprecher
Prof. Dr. Rainer Walden †	2009 - 2015

Als assistierende Mitarbeiterinnen im AMMO-Sekretariat waren tätig:

Dipl.-Math. Tatiana Schenck	2009 - 2012
Dipl.-Math. Natalja Zoludev	2013 - 2015

Wie bereits im Vorwort angesprochen, ist der Hauptzweck dieses Zusammenschlusses, das spezifische Wissen und die Erfahrungen der einzelnen Wissenschaftlerinnen und Wissenschaftler, die ihre Forschungsausrichtungen auf unterschiedlichen Gebieten der Mathematik haben und weitreichende Kenntnisse aus verschiedenen Praxisumfeldern mitbringen, im gemeinsamen Schnittbereich der mathematischen Modellierung und Optimierung zu bündeln und aus diesem Kernbereich gemeinsame Forschungs- und Entwicklungsprojekte zu generieren.

Die mathematische Modellierung, Optimierung und Simulation stellen heute unverzichtbare Werkzeuge bei der Entwicklung und Analyse von neuen Produkten oder Produktionsverfahren dar. Um beispielsweise herauszufinden, wie ein Kraftwerk effektiv Energie produzieren kann (d.h. mit niedrigeren Kosten und geringerem Ausstoß von Schadstoffen), braucht man nicht die Tausende von Kubikmetern des Brennstoffes zu verarbeiten, damit die entsprechenden Messungen durchgeführt werden können. Mit Hilfe von mathematischen Methoden kann man den Prozess der Energieproduktion simulieren und eine optimale Einstellung für die Parameter berechnen, die es erlaubt, die Produktionskosten deutlich zu senken.

Die große Anzahl und Vielseitigkeit der Methoden zur mathematischen Modellierung und Optimierung ermöglicht ihre Anwendung bei den Problemstellungen aus den unterschiedlichsten Bereichen der Wirtschaft und des öffentlichen Lebens. Das Wissenschaftlerteam des Forschungs- und Entwicklungsschwerpunktes *Angewandte Mathematische Modellierung und Optimierung* (FSP AMMO) verfügt über ein breites Spektrum an Erfahrung in der Anwendung von Modellierungs- und Optimierungsstrategien. Die persönlichen Forschungsschwerpunkte der AMMO-Mitglieder kann man im

Wesentlichen den folgenden Fachgebieten zuschreiben: Finanz- und Versicherungsmathematik, Risikomanagement, Simulation betrieblicher Prozesse, Entwicklung von Entscheidungsunterstützungs- und Optimierungssystemen, Multidisziplinäre Modellierung und Simulation z.B. mit Modelica, Angewandte Strukturoptimierung, Etablierung, Weiterentwicklung und Nutzung von Modellen der Systembiologie, Softwareentwicklung und Prozessdatenverarbeitung.

Im FSP AMMO werden die Kompetenzen auf verschiedenen Anwendungsgebieten der mathematischen Modellierung und Optimierung gebündelt, d.h. es werden mathematische Methoden für die unterschiedlichen Anwendungen zusammengeführt, um ein beachtliches Repertoire an Lösungsmöglichkeiten für aktuelle Problemstellungen bereitzustellen. So kann beispielsweise eine Fragestellung aus der Logistik durch eine geeignete Kombination einer herkömmlichen Lösungsmethode mit Optimierungsmethoden anderer Fachdisziplinen eventuell besser oder schneller gelöst werden als nur mit der konventionellen Methode. Umgekehrt besteht auch die Möglichkeit der Verwendbarkeit einer speziellen Anwendungsmethode für verschiedene Problemstellungen.

Die Tätigkeit des FSP AMMO wird durch enge Kooperationen mit anderen Forschungseinrichtungen und Unternehmen ergänzt. Diese Zusammenarbeit reicht von der gemeinsamen Betreuung von Praxissemester und Abschlussarbeiten bis zur Bearbeitung umfangreicher Projekte auf EU-Ebene. Seit der Gründung von AMMO bis heute sind viele unterschiedliche Projekte initiiert und durchgeführt worden.¹

Es ist die Veröffentlichungsreihe „AMMO – Berichte aus Forschung und Technologietransfer“ (ISSN 2198-4824) entstanden. In dieser Reihe erschienen bislang die folgenden Beiträge²:

Nr.	Titel	Autoren	Jahr
1	Informationen über den Forschungs- und Entwicklungsschwerpunkt Angewandte Mathematische Modellierung und Optimierung	AMMO-Team	12/2013
2	Methoden der quantitativen rechnergestützten CTG-Analyse	R. Walden, V. M. Roemer	04/2014
3	Entwicklung einer Dispatcherfunktion zur Überprüfung von Nominierungsmengen in der Betriebsführung von Erdgasspeichern	R. Ueckerdt, H.-W. Schmidt, M. Weber, E. Mindlina	06/2014
4	Diskrete Modellierung und Optimierung praxisrelevanter Prozesse mit Petri-Netzen	S. Proß	09/2014
5	Optimumgraphen	H.-J. Kruse	10/2014
6	Lösungsansätze für Konfliktsituationen bei Feuerprozessen in kapazitierten Petri-Netzen	T. Kleine-Döpke, H.-J. Kruse	06/2015
7	Formalismen für gefärbte Petri-Netze und Verfahren zur effizienten Berechnung von aktiven Modus-Mengen	J. Silberberg, T. Lask, B. Bachmann	Anfang 2016

¹Hierzu sei auf den Link <http://www.fh-bielefeld.de/ammo/projekte> verwiesen.

²Diese Beiträge sind als pdf-Dokumente über den Link <http://www.fh-bielefeld.de/ammo/veroeffentlichungen/ammo-berichte-aus-forschung-und-technologietransfer> zu beziehen.

Wer und Was ist AMMO?

Zudem hat sich eine Vortragsreihe „Treffpunkt AMMO“ etabliert, zu der Wissenschaftlerinnen und Wissenschaftler von anderen Hochschulen sowie Vertreterinnen und Vertreter von Unternehmen, insbesondere aber auch Studierende eingeladen werden, um je nach Thematik sowohl über aktuelle Forschungs- und Entwicklungsergebnisse als auch über konkrete Problemstellungen aus der Praxis zu informieren und zu diskutieren.¹

Zu guter Letzt soll nicht unerwähnt bleiben, dass AMMO schon mehrmals mit einem Informationsstand auf der Cebit-Ausstellung in Hannover vertreten war (2011-2013), um eine breite Öffentlichkeit über die Forschungs- und Entwicklungstätigkeiten von AMMO zu informieren.

Dezember 2015

Sprecherin des FSP AMMO

¹<http://www.fh-bielefeld.de/ammo/veranstaltungen/veranstaltungsreihe-treffpunkt-ammo>

Die Autoren

Beteiligte AMMO-Mitglieder



Prof. Dr. phil. Bernhard Bachmann

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

bernhard.bachmann@fh-bielefeld.de

Fachgebiete: Numerische Mathematik, Optimierung, Symbolische und numerische Behandlung großer hybrider differential-algebraischer Gleichungen

Seit 1999 als Professor für Mathematik und ihre technischen Anwendungen an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Gründungsmitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld. Gründungsmitglied der *Modelica Association*. Gründungs- und Vorstandsmitglied des *Open Source Modelica Consortium* (OSMC). Nationale und internationale F&E-Projekte im Bereich der Modellierung, Simulation und Optimierung hybrider dynamischer Systeme.



Prof. Dr. math. Friedrich Biegler-König

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

friedrich.biegler-koenig@fh-bielefeld.de

Fachgebiete: Numerische Mathematik, Künstliche Intelligenz

Seit 1996 als Professor für Angewandte Mathematik und Informatik an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Gründungsmitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld. F&E-Projekte in den Bereichen bionische Optimierungsverfahren und Neuronale Netze.



Prof. Dr. rer. nat. Claudia Cottin

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

claudia.cottin@fh-bielefeld.de

Fachgebiete: Finanz- und Versicherungsmathematik, Aktuarwissenschaften, Quantitatives Risikomanagement

Seit 1997 als Professorin für Finanz- und Versicherungsmathematik an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* sowie im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Gründungsmitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld. F&E-Projekte zu diversen aktuariellen Fragestellungen und im Bereich der stochastischen Modellbildung und Simulation.



Prof. Dr. rer. pol. Hermann-Josef Kruse

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

hermann-josef.kruse@fh-bielefeld.de

Fachgebiete: Wirtschaftsmathematik, Operations Research

Seit 1995 als Professor für Wirtschaftsmathematik (insbesondere Operations Research) an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Gründungsmitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld. F&E-Projekte im Bereich der Optimierung und Simulation diskreter Systeme zur Entscheidungsunterstützung bei betrieblichen Problemstellungen.



Timo Lask (M.Sc.), geb. Kleine-Döpke

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

timo.lask@fh-bielefeld.de

Fachgebiete: Petri-Netze, Personalplanung, Echtzeitsysteme

Hat sein Master-Studium im Fach *Optimierung und Simulation* und sein Bachelor-Studium in der *Angewandten Mathematik* an der Fachhochschule Bielefeld absolviert. Arbeitete seit der Mitte seines Bachelor-Studiums als SHK im FSP-AMMO-Projekt „Coloured Petri Nets“ (CPN) von Prof. Dr. Hermann-Josef Kruse und hat sich in den letzten Jahren unter anderem in dieser Fachrichtung spezialisiert. Seit Ende 2014 an der Fachhochschule Bielefeld als wissenschaftlicher Mitarbeiter in Forschungsprojekten beschäftigt und seit Januar 2015 ein festes AMMO-Mitglied.

Die Autoren



Prof. Dr. math. Svetozara Petrova

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

svetozara.petrova@fh-bielefeld.de

Fachgebiete: Numerische Mathematik, Optimierung, Interpolation und Approximation

Seit 2008 als Professorin für Angewandte mathematische Modellierung und Optimierung an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Mitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld seit 2009. F&E-Projekte in den Bereichen der Strukturoptimierung und Mehrskalprobleme.



Dr. rer. nat. Sabrina Proß

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

sabrina.pross@fh-bielefeld.de

Fachgebiete: Mathematische Modellierung und Optimierung biologischer Prozesse, hybride Petri Netze

Seit 2012 als Lehrkraft für Ingenieurmathematik und Statistik an der Fachhochschule Bielefeld tätig. Mitglied im Forschungsschwerpunkt *Angewandte Mathematische Modellierung und Optimierung* (FSP AMMO) seit 2009. Promotion im Bereich der hybriden Modellierung und Optimierung von biologischen Prozessen. Spezialisierung auf die Modellierung mit Petri-Netzen.

Weitere Autoren



Dipl.-Wirt.Math. Ralf Derdau

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

ralf.derdau@fh-bielefeld.de

Fachgebiete: Wirtschaftsmathematik, Informatik

Seit 1994 als wissenschaftlicher Mitarbeiter an der Fachhochschule Bielefeld tätig und dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung und Simulation* des Fachbereichs *Ingenieurwissenschaften und Mathematik*. Aufgabengebiete sind die Bereitstellung und der Support der informationstechnischen Infrastruktur des Studienganges *Angewandte Mathematik*, sowie die Schulung dort eingesetzter Mathematik- und Simulationssoftware.



Dimitri Kriger (B. Sc.)

ELA Container GmbH

Münster

dimakriger@web.de

Fachgebiete: Logistik

Hat sein Bachelor-Studium in der *Angewandten Mathematik* an der Fachhochschule Bielefeld absolviert. Spezialisierte sich im Laufe des Studiums auf das Gebiet der Logistik. Ist nun bei Ela [Container] als Logistiker angestellt.

Die Autoren



Dr. rer. nat. Andreas Kronshage

LWL-Museum für Naturkunde
Außenstelle Heiliges Meer
Andreas.Kronshage@lwl.org

Fachgebiete: Gewässerökologie, Fauna (insbes. Amphibien und Reptilien, Libellen, Heuschrecken, Tagfalter, Vögel) und Flora, ökologische Landesforschung (Westfalen)

Geograf und Landschaftsökologe, seit 1994 freiberuflicher Gutachter vor allem im Bereich Umweltplanung und Tierökologie, Mitarbeit an landesweiten und regionalen faunistischen und floristischen Erfassungsprojekten, 1999 bis 2004 wissenschaftlicher Mitarbeiter im Biologischen Institut Metelen / Forschungseinrichtung für Biotop- und Artenschutz, seit 2005 wissenschaftlicher Mitarbeiter in der Außenstelle Heiliges Meer des LWL-Museums für Naturkunde in den Bereichen Umweltbildung, Durchführung und Entwicklung von biologisch-ökologischen Kursen, Seminaren, Exkursionen und Fachtagungen, Öffentlichkeitsarbeit und arten- und naturschutzorientierte Forschung.



Natalie Spent (M. Sc.)

HiQ Projects GmbH
Ehrenstraße 13, 50672 Köln
natalie.spent@hiq-projects.com

Fachgebiete: Logistikprobleme, Operations Research

Absolvierte ihr Bachelor-Studium in *Angewandte Mathematik* und ihr Master-Studium in *Optimierung und Simulation* an der Fachhochschule Bielefeld. Spezialisierte sich im Laufe des Studiums auf das Gebiet der kombinatorischen Optimierung, insbesondere Entsorgungs- bzw. Transportprobleme. Ist nun bei HiQ Projects als Beraterin für Software-Qualitätssicherung tätig.

Bisher in der Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik erschienen:

- Band 1:
„Hybrid Modeling and Optimization of Biological Processes“
Sabrina Pross (Juli 2013)
- Band 2:
„Petri-Netz-Formalismen und Lösungsansätze für allgemeine Konfliktsituationen bei Feuerprozessen in Petri-Netz-Modellen“
Bernhard Bachmann, Timo Kleine-Döpke, Hermann-Josef Kruse, Lennart Ochel, Sabrina Proß (November 2014)
- Band 3:
„Application Techniques of Endophytes“
Desiree Jakobs-Schönwandt, Matthias Döring and Anant Patel (eds.), Juli 2015

Kontakt:

Prof. Dr.-Ing. Rolf Naumann
Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und Mathematik
Interaktion 1
33619 Bielefeld
Telefon +49.521.106-7483
rolf.naumann@fh-bielefeld.de



**Forschungsreihe des Fachbereichs
Ingenieurwissenschaften und Mathematik**

Herausgeber

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften
und Mathematik

Band 4

Februar 2016

ISSN 2196-6192