

Fachbereich
Ingenieurwissenschaften
und Mathematik
Angewandte Mathematik

Theoretische Grundlagen zur Implementierung von Lösungsstrategien für das Laubharkproblem

Hermann-Josef Kruse

März 2018



FH Bielefeld
University of
Applied Sciences

Dieses Manuskript ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Skriptes oder Teilen daraus, sind vorbehalten. Kein Teil des Skriptes darf ohne schriftliche Genehmigung der Autoren in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert werden oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2018 Prof. Dr. Hermann-Josef Kruse

Vorwort

In den letzten Jahren wurden im Rahmen einer Reihe von Qualifizierungsarbeiten (Bachelor-, Master- und Projektarbeiten) im Bachelor-Studiengang *Angewandte Mathematik* sowie im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik* der Fachhochschule Bielefeld viele spezielle Untersuchungen an der Alltagsproblematik des jährlichen Laubharkens in Gärten oder Parks als ein spezifisch ausgeprägter Entsorgungsprozess vorgenommen. Hierbei wurden für unterschiedliche Fragestellungen spezifische Lösungsansätze entwickelt, die Eingang in ein Matlab-basiertes Programm („Laubharktool LHT“) finden sollen. Mit diesem Tool sollen schließlich gezielte Effizienzanalysen vorgenommen werden können, um einem Anwender für analoge Problemausprägungen geeignete Lösungsverfahren anbieten zu können.

In diesem Manuskript werden die theoretischen Grundlagen zusammengefasst, um die Implementierung der Lösungsansätze zu erleichtern, indem ein einheitlicher Formalismus vorgegeben wird und die Berechnungen der maßgeblichen Aufwände des gesamten Entsorgungsprozesses (Aufwände für Harken, Abtransportieren, unproduktive Wege) anhand von algorithmischen Beschreibungen angegeben werden. Zudem werden einige Harkstrategien vorgeschlagen (ebenfalls mit Pseudocodes).



Inhaltsverzeichnis

1	Einleitung	1
2	Das Laubharkproblem als kombinatorisches Optimierungsproblem	3
2.1	Gartenmodelle	3
2.2	Aufwandsberechnung	8
2.2.1	Berechnung des Harkaufwandes	8
2.2.2	Darstellung der Harkprozesse durch Nachfolgerfunktionen	13
2.2.3	Berechnung von unproduktiven Wegezeiten	17
2.2.4	Berechnung des Transportaufwandes	20
2.3	Das formale Laubharkmodell	22
3	Lösungsstrategien für das Laubharkproblem	25
3.1	Hubbestimmung in einem Cluster	25
3.1.1	Hubzentrierung in einem Cluster	26
3.1.2	Hubausrichtung in einem Cluster	27
3.1.3	Sonderbehandlung von Leerfeldern	27
3.2	Konstruktionsverfahren	28
3.2.1	Zickzack-Strategien	28
3.2.2	Clusterorientiertes Konstruktionsverfahren	30
3.2.3	Huborientiertes Konstruktionsverfahren	30
3.3	Verbesserungsverfahren	33
3.3.1	Lösungsverbesserung durch einfachen Nachfolgertausch inner- halb eines Clusters	33
3.3.2	Lösungsverbesserung durch mehrfachen Nachfolgertausch inner- halb eines Clusters	34
3.3.3	Lösungsverbesserung durch Einfügestrategie	34
3.3.4	Lösungsverbesserung durch Clustervereinigung	35
	Literaturverzeichnis	37
A	Glossar	39
B	Algorithmen	42

C	Lösungsstrategien am Beispiel	58
C.1	Zickzack-Strategien am Beispiel	59
C.1.1	Einfache Zickzack-Strategie	59
C.1.2	Unterteilte Zickzack-Strategie	60
C.2	Clusterorientiertes Konstruktionsverfahren am Beispiel	62
C.3	Huborientiertes Konstruktionsverfahren am Beispiel	63
C.3.1	Huborientiertes Konstruktionsverfahren via Tiefensuche	63
C.3.2	Huborientiertes Konstruktionsverfahren via Breitensuche	66
C.4	Abschließende Erläuterungen zu den Beispielrechnungen	66

1 Einleitung

Das jährliche Laubharken in Gärten oder Parks stellt eine sehr spezielle, alltagsnahe Ausprägung von allgemeinen Entsorgungsprozessen dar und eignet sich nicht zuletzt aufgrund dieses „Bekanntheitsgrades“ als didaktisches Musterbeispiel, um in die Bereiche der mathematischen Modellierung, Optimierung und Simulation einzuführen, zumal es sich in der Gesamtheit als ein recht komplexes Problem herausstellt. Schließlich ergeben sich hierbei mehrere Fragestellungen, die simultan oder sukzessiv zu lösen sind:

- Soll das zu entsorgende Laub auf viele kleine oder wenige große Laubhaufen zusammengeharkt werden?
- Wo sollen diese Laubhaufen gebildet werden?
- Von welchen Feldern soll zu welchen Laubhaufen geharkt werden?
- In welcher Reihenfolge sollen dabei die einzelnen Felder bearbeitet werden?
- Welcher Aufwand ergibt sich dabei einerseits beim Harkprozess, andererseits beim Transportprozess (Abtransport der Laubhaufen mit einem Laubwagen zu einer oder zu mehreren Kompoststellen).
- Wie lassen sich dabei unproduktive Wege möglichst vermeiden?

Die Bestimmung von optimalen Standorten für die Laubhaufen lässt sich auch als ein spezielles Hub-Location-Problem auffassen. Somit ist es nicht verwunderlich, wenn gängige Lösungsansätze für Standortplanungsprobleme in adaptierter Form auch zur Lösung des Laubharkproblems zur Anwendung kommen.

Einen Vorschlag für die Modellierung des Laubharkproblems findet man in [5]. Hierbei wird ein Garten durch beliebig feine Rasterung in Matrixform gebracht, wodurch endlich viele Felder entstehen und die Grundlage für ein kombinatorisches Optimierungsmodell gegeben ist. Durch die Festlegung der Nachbarschaft dieser Felder, welche die Möglichkeiten von direkten Harkvorgängen beschreiben, ergibt sich auf kanonische Weise ein ungerichteter Graph, dessen Kantenbewertungen den zeitlichen Aufwand sowohl für den Hark- als auch für den Transportprozess angeben. Die Minimierung des Gesamtaufwandes, in den Einzelaufwände für die Teilprozesse parametrisiert eingehen,

1 Einleitung

lässt sich somit als ein kombinatorisches Optimierungsproblem formulieren (Näheres hierzu siehe in Kapitel 2).

Für diejenigen Leser/innen, die sich nicht so sehr für eine anschauliche und ausführliche Herleitung der Modellierung des Laubharkprozesses als ein kombinatorischen Optimierungsproblems interessieren, mag ein direkter Einstieg zum Ende von Kapitel 2 genügen, wo das mathematische Optimierungsmodell in konzentrierter Form vorgestellt wird (Abschn. 2.3).

Einige Lösungsansätze für dieses spezielle Optimierungsproblem findet man in [6]. Diese basieren zum Teil auf klassischen Heuristiken für kombinatorische Optimierungsprobleme und sind auf die spezielle Problematik passend zugeschnitten (hierzu siehe Kapitel 3).

2 Das Laubharkproblem als kombinatorisches Optimierungsproblem

Um ein reales Laubharkproblem mit Hilfe von mathematischen Optimierungsverfahren einer Lösung zuführen zu können, bedarf es zunächst einer Abstraktion des Problems zwecks Bildung eines mathematischen Modells. Die Modellbildung erfolgt in mehreren Stufen. Zunächst wird ein abstraktes Modell für den Garten geschaffen. Anschließend werden Überlegungen zur Festlegung einer geeigneten Zielgröße angestellt und ein Aufwandsmodell entwickelt. Dabei wird die Relevanz von problemspezifischen Nebenbedingungen und Annahmen diskutiert.

2.1 Gartenmodelle

In der folgenden Abbildung ist ein kleiner Garten skizziert. Man erkennt die Rasenfläche (hellgrün), den Teich (blau), das Gerätehaus (rot), die Kompoststelle (dunkeloliv), einige Bäume (braun) und die gepflasterten Gehwege (grau).¹

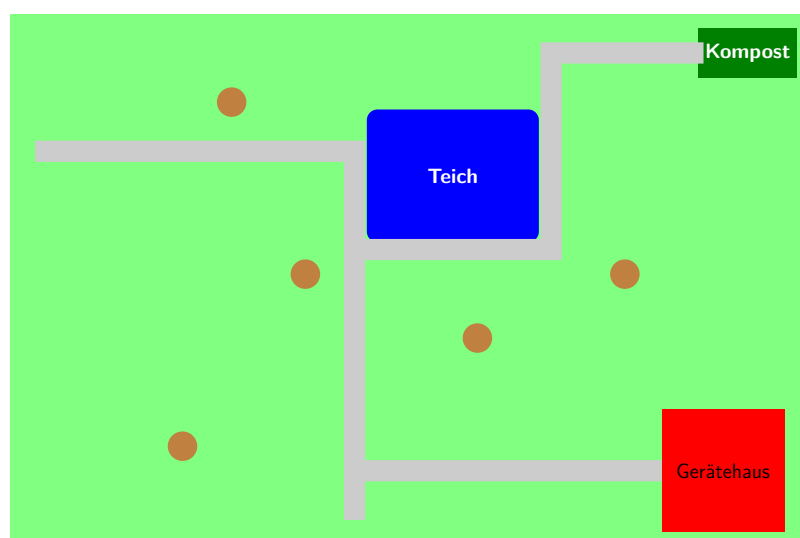


Abbildung 1: Skizze eines Gartens

¹ Natürlich sind Rasenflächen nicht in allen Gärten rechteckig, aber diese Gegebenheit wird die Modellierung des Gartens und die Veranschaulichung der noch zu entwickelnden Laubharkstrategien erleichtern; daher werde die Rechteckform o.B.d.A. angenommen.

Als weitergehende Abstraktionsstufe² bietet sich die Rasterung der Rechteckfläche an (siehe Abb. 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 ($i = 1, \dots, m, j = 1, \dots, n$).³

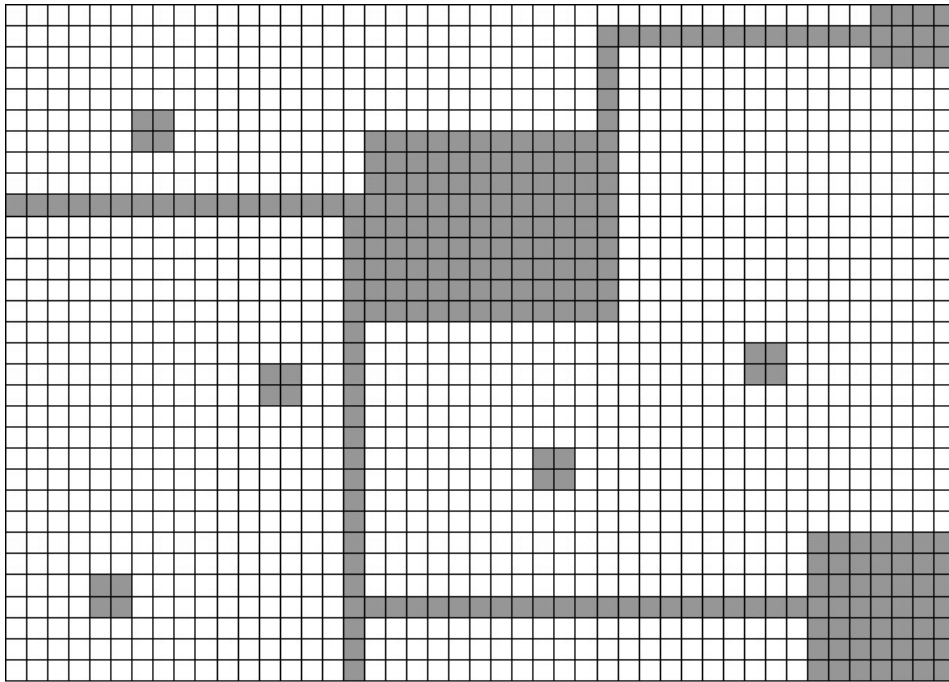


Abbildung 2: Gerasterte Gartendarstellung

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

$$a = (i - 1) \cdot n + j. \quad (2.1)$$

Im Garten wird das Laub nun in der Regel sehr unterschiedlich auf der Rasenfläche (und den Gehwegen) verteilt liegen.⁴ 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,

² Als 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.

³ Wie fein- bzw. grobmaschig die Rasterdarstellung sein soll, bleibt dem Modellbauer überlassen. 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.

⁴ Hierbei 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.

wonach sich das Laubharkproblem nicht mehr stellte. Daher werden die Laubmengen in den Feldern höchstens nach Augenschein geschätzt und registriert. Jedes Feld a bekommt somit eine (wie auch immer ermittelte) *Laubmenge* $M(a)$ zugeordnet.⁵ Diese Laubmenge wird in *Laubmengeneinheiten* [ME] gemessen.⁶ 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 vornehmlich die ganzzahlige Variante benutzt 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. 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 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 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 unter-

⁵ Die Laubmenge $M(a)$ für ein Feld $a = (i, j)$ wird bei Koordinatendarstellung mit $M(i, j)$ bezeichnet.

⁶ Hierbei 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.

scheiden zu können (und 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 gilt:

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

Hierbei ist allerdings noch zu vereinbaren, was unter der „Nachbarschaft“ zwischen Feldern konkret verstanden werden soll. Hierzu bieten sich zwei Nachbarschaftsmodelle an.⁷

Zwei Felder a und b heißen *benachbart*, wenn gilt⁸:

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

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. 4 links). Die nächste Variante (b) lässt zudem noch diagonales Harken und somit vier weitere Möglichkeiten zu (vgl. Abb. 4 rechts):

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

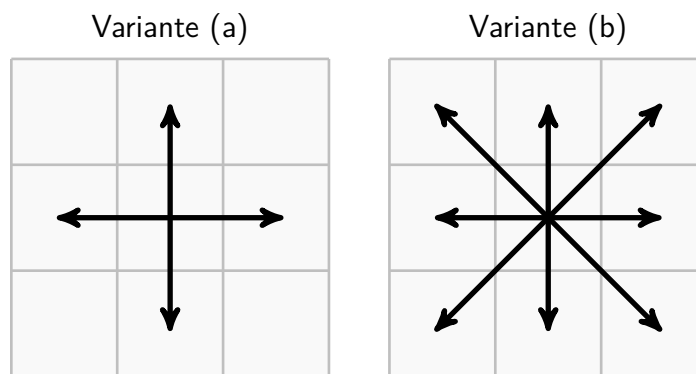


Abbildung 4: Nachbarschaftsvarianten

⁷ Ein weiteres interessantes Nachbarschaftsmodell wäre die Darstellung des Gartens als Wabenmuster, sodass sich auf kanonische Weise jeweils bis zu sechs Feldnachbarn ergeben. Diese Darstellungsmöglichkeit wird im Folgenden aber nicht weiter ausgeführt, zumal diese sich in die Graphenmodellen wiederfinden lässt.

⁸ Für „Randfelder“ gelten auf kanonische Weise eingeschränkte Bedingungen.

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 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. 3 vorgestellte Gartenbeispiel in Matrixform erhält je nach Nachbarschaftsvariante die folgenden Darstellungen als Graph (siehe Abb. 5 bzw. Abb. 6), an denen unschwer zu erkennen ist, dass somit auch beliebige Gartenformen mittels Graphen abstrahiert werden können.

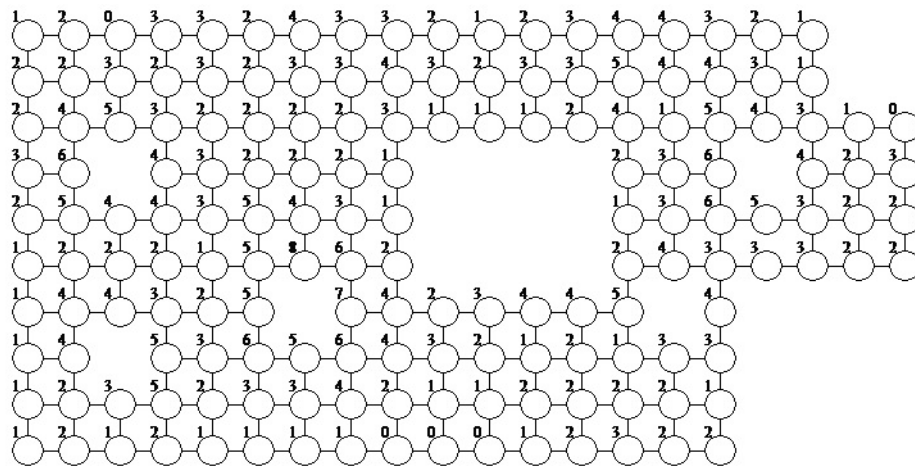


Abbildung 5: Graphendarstellung eines Gartens gemäß Nachbarschaftsvariante (a)

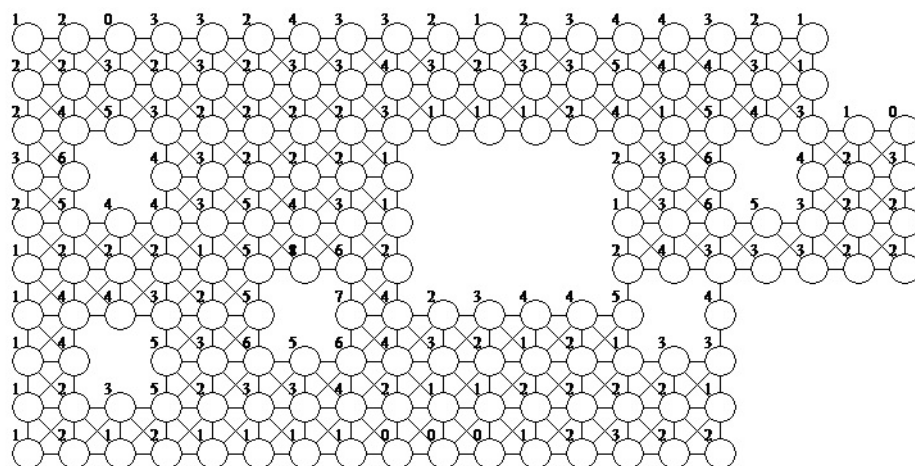


Abbildung 6: Graphendarstellung eines Gartens gemäß Nachbarschaftsvariante (b)

Zudem bietet es sich an, auch die Kanten des Graphen zu bewerten, wobei die Elemente $c_{a,b}$ der *Bewertungsmatrix* C Feldabstände oder Harkaufwände zwischen benachbarten Knoten (Feldern) a und b wiedergeben können.⁹ Die Darstellung eines Gartens als knoten- und kantenbewerteter Graph hat den beträchtlichen Vorteil, dass auszulassende Felder nicht mehr in der Knotenmenge enthalten sind. Zudem lässt sich aus der Bewertungsmatrix C des Graphen sehr einfach eine Entfernungsmatrix D ermitteln, mit deren Hilfe die Hark- und Transportaufwände berechnet werden können, sobald die Hark- und Transportwege bestimmt sind.

2.2 Aufwandsberechnung

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 dieser legt 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, so dass 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.

2.2.1 Berechnung des Harkaufwandes

Der Harkprozess besteht aus dem Laubharken selbst, also aus Tätigkeiten, bei denen das Laub bewegt wird, aber auch aus unproduktiven Wegezeiten, die zwischen den einzelnen Harkvorgängen anfallen können. Diese werden im Folgenden getrennt voneinander betrachtet. Zunächst wird in diesem Abschnitt der eigentliche Harkprozess betrachtet; die unproduktiven Wege werden in Abschn. 2.2.3 untersucht.

Eine sehr einfache Berechnungsvorschrift ist es, den *Harkaufwand* $HA(a, b)$ eines

⁹ Näheres hierzu in Abschn. 2.2

Teilprozesses, d.h. den Aufwand für das Harken der aktuellen 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 zunächst 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. Mit $M^*(a)$ ist diejenige aktuelle Laubmenge im Feld a gemeint, die zum Zeitpunkt des Harkens von Feld a in das Nachbarfeld b vorliegt; die aktuelle Laubmenge $M^*(a)$ wird sich durch den ausgeführten Harkprozess im Allgemeinen von der initialen Laubmenge $M(a)$ unterscheiden.

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 sein¹⁰:

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

Somit ergibt sich (2.6) auf kanonische Weise als Spezialfall von (2.7), wenn $\alpha_H(a, b) \equiv \alpha_H$ für alle benachbarten 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 aktuelle Laubmenge $M^*(a)$

¹⁰ Die Harkaufwandsfaktoren $\alpha_H(a, b)$ können beim Graphenmodell ohne Weiteres als Elemente der Bewertungsmatrix C hinterlegt werden.

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 vorliegender 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 ändert¹¹:

$$HA(a, b) = \left\lceil \frac{M^*(a)}{M_k} \right\rceil \cdot \alpha_H(a, b). \quad (2.8)$$

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 Laubharkmenge 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.

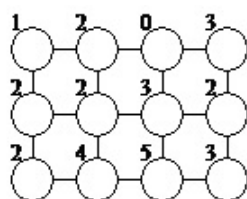


Abbildung 7: Nordwestlicher Ausschnitt aus Abb. 5

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 Laubharkmenge 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

¹¹ Hierbei sind $\lceil \cdot \rceil$ die Gaußklammern für die Aufrundung zur nächstgrößeren ganzen Zahl.

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. 8 als intuitives „Zickzack-Harken“ skizziert ist.

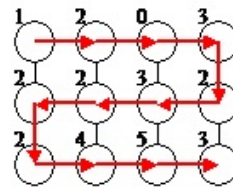


Abbildung 8: Zickzack-Harken (Variante 1)

Der gesamte Harkaufwand, der sich aus den in Tab. 1 aufgelisteten Einzelaufwänden (Anzahl der Harkzüge) zusammensetzt, beträgt dabei 124 Harkzüge.

Tabelle 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

2 Das Laubharkproblem als kombinatorisches Optimierungsproblem

Eine alternative „Zickzack-Strategie“ ist in Abb. 9 skizziert. Der gesamte Harkaufwand für diese Alternative beträgt 150 Harkzüge (vgl. Tab. 2).

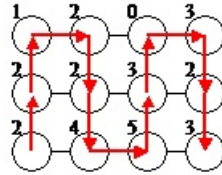


Abbildung 9: Zickzack-Harken (Variante 2)

Tabelle 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. 10 eine weitere Harkmöglichkeit gegenübergestellt. Der gesamte Harkaufwand für diese 3. Variante beträgt 47 Harkzüge (vgl. Tab. 3).

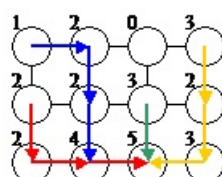


Abbildung 10: Hark-Variante 3

Tabelle 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.¹² Zudem sei bereits an dieser Stelle angemerkt, dass bei der Harkvariante 3 unproduktive Wege während des Harkprozesses entstehen, zumal dieser mehrfach unterbrochen werden muss, und zwar jeweils beim „Farbwechsel“ der Teilharkprozesse; diese unproduktiven Wege nehmen Zeit in Anspruch und sind somit dem produktiven Harkaufwand zuzuschlagen. Demgegenüber treten bei den Zickzack-Strategien keine solchen unproduktiven Wege auf. Näheres hierzu wird noch in Abschn. 2.2.3 diskutiert werden.

2.2.2 Darstellung der Harkprozesse durch Nachfolgerfunktionen

Um später zur Entwicklung von effizienten Strategien auf Verfahren aus der kombinatorischen Optimierung zurückgreifen zu können, wird vorweg ein Darstellungsmodell für Harkprozesse eingeführt, an welchem die Berechnung des jeweiligen Harkaufwands in einheitlicher Form vollzogen werden kann.

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 Nachbarmfeld $b = s(a) \in V$ zugewiesen, wohin die aktuelle Laubmenge $M^*(a)$ geharkt wird. Eine anschauliche Darstellung einer Harkvorschrift ist die Tabellenform (vgl. Abb. 10):

¹² In größeren Gärten bestimmen auch noch weitere Kenngrößen die Suche nach effizienten Laubharkstrategien, z.B. die Anzahl und die Größe der zu errichtenden Laubhaufen.

Tabelle 4: Beispiel einer Harkvorschrift mittels 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 Tabelle 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 Haufenfeld 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; es stellt somit ein *isoliertes Haufenfeld* dar.

Damit eine Abbildung $s : V \rightarrow V$ eine Harkvorschrift darstellt, wobei ausschließlich entlang von Kanten des Graphen G geharkt wird, muss für jeden Knoten $a \in V$ die *Nachbarschaftsbedingung* gelten:

$$a \neq s(a) \Rightarrow [a, s(a)] \in E. \quad (2.9)$$

Man mache sich klar, dass durch eine Abbildung $s : V \rightarrow V$, die der Nachbarschaftsbedingung genügt, 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. Um ein sinnvolles Harken zu gewährleisten, sollte dieser Digraph \vec{G}_s zudem zyklensfrei sein.

Eine solche Abbildung $s : V \rightarrow V$, die sowohl der Nachbarschaftsbedingung (2.9) genügt als auch einen zyklensfreien Digraph $\vec{G}_s = \langle V, \vec{E}_s \rangle$ mit $\vec{E}_s = \{ \langle a, s(a) \rangle \mid a \in V \wedge a \neq s(a) \}$ induziert, wird im Folgenden *Nachfolgerfunktion* genannt.¹³

Eine Harkvorschrift in Form einer Nachfolgerfunktion stellt auf kanonische Weise eine Lösung des Laubharkproblems dar, denn es werden die generierten Laubhaufen hinsichtlich der Anzahl, Lage und kumulierten Laubmenge eindeutig festgelegt. Zur Überprüfung, ob eine Abbildung $s : V \rightarrow V$ eine Nachfolgerfunktion ist, kann ein einfaches Prüfmodul implementiert werden (hierzu siehe Algorithmus 1 im Anhang B).

Man mache sich ebenfalls klar, dass der durch eine Nachfolgerfunktion s induzierte Digraph \vec{G}_s eine „inverse Baumstruktur“ besitzt, d.h. der inverse Digraph ist ein gerichteter Wald. Im obigen Beispiel ergibt sich ein gerichteter Wald mit zwei Wurzelbäumen,

¹³ Vgl. [5], S. 82.

wobei der eine Baum den Wurzelknoten 11 und die Blätter 1, 4, 5 und 7 hat, der zweite Baum allein aus dem isolierten Knoten 3 besteht (siehe Abb. 11).

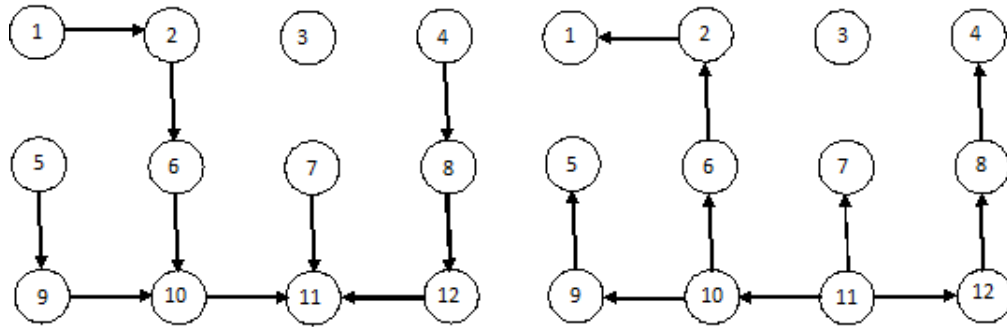


Abbildung 11: Durch Nachfolgerfunktion induzierter Digraph (links) und Wald (rechts)

Die Quellen¹⁴ des durch eine Nachfolgerfunktion s induzierten Digraphen \vec{G}_s werden im Folgenden auch *Harkquellen* genannt; sie bilden die *Harkquellenmenge* $Q(s)$ von s . Die Senken¹⁵ von \vec{G}_s stellen die Haufenfelder (Hubs) des Harkprozesses dar; sie bilden die *Hubmenge* $Hub(s)$ von s .

Um zu gewährleisten, dass eine Nachfolgerfunktion s nicht nur im Initialzustand, sondern auch während des gesamten Harkprozesses vorgegebene Kapazitätsbeschränkungen $\overline{M}(a)$ für alle Felder $a \in V$ erfüllt, wird noch ein Zulässigkeitsbegriff eingeführt: Eine Nachfolgerfunktion s wird *zulässig* bzgl. \overline{M} genannt, wenn bei dem durch s induzierten Harkprozess beachtet wird, dass in keinem Feld a und erst recht in keinem Haufenfeld mehr Laub angehäuft wird, als gemäß maximaler Laubmenge $\overline{M}(a)$ bzw. \overline{M} erlaubt ist. Um diesen Sachverhalt zu formalisieren, wird zunächst zu jedem Knoten a des Digraphen \vec{G}_s die *Erreichbarkeitsmenge* $R(a)$ bestimmt; diese enthält alle Knoten $i \in V$, von denen aus der Knoten a entlang einer Pfeilfolge in \vec{G}_s erreichbar ist.¹⁶ Die Abbildung $M_s^* : V \rightarrow \mathbb{R}_{\geq}$ mit der Bildungsvorschrift

$$M_s^*(a) := \sum_{k \in R(a)} M(k) \text{ für alle Knoten } a \in V \quad (2.10)$$

heißt die durch s induzierte *Anhäufungsfunktion*. Eine Nachfolgerfunktion s heißt dann *zulässig* bzgl. \overline{M} , wenn gilt¹⁷:

$$M_s^*(a) \leq \overline{M}(a) \text{ für alle Knoten } a \in V. \quad (2.11)$$

Es ist leicht einzusehen, dass im konstanten Falle (d.h. $\overline{M}(a) = \overline{M}$ für alle $a \in V$) die Zulässigkeitsprüfung allein auf die Hubs $a \in Hub(s)$ eingeschränkt werden kann,

¹⁴ Quellen sind Knoten ohne Pfeileingänge in einem Digraph.

¹⁵ Senken sind Knoten ohne Pfeilausgänge in einem Digraph.

¹⁶ Zur Bestimmung einer Erreichbarkeitsmenge $R(a)$ dient der Algorithmus 3 im Anhang B.

¹⁷ Für eine Zulässigkeitsprüfung einer Nachfolgerfunktion dient der Algorithmus 2 im Anhang B.

zumal sich daraus die Zulässigkeit für alle Knoten ergibt. Umgekehrt (d.h. $\overline{M}(a)$ nicht konstant) kann es Fälle geben, dass die Bedingung (2.11) zwar für alle Hubs, nicht aber für alle sonstigen Knoten gelten muss.

Zur effizienten Bestimmung des Harkaufwandes¹⁸ anhand von (zulässigen) Nachfolgerfunktionen wird eine einfache Rechenprozedur benutzt¹⁹: Zu gegebener (zulässiger) Nachfolgerfunktion s eines Graphen $G = [V, E]$ wird die Anhäufungsfunktion M_s^* ermittelt. Danach wird zu jedem Nicht-Hub $a \in V \setminus \text{Hub}(s)$ der Harkaufwand $HA(a, s(a))$ gemäß einer der Harkaufwandsformeln (2.6) - (2.8) bestimmt. Abschließend werden die Einzelaufwände aufsummiert:

$$HA_{\Sigma}(s) = \sum_{a \in V \setminus \text{Hub}(s)} HA(a, s(a)). \quad (2.12)$$

Für die Berechnung des Harkaufwandes einer Harkvorschrift ist die Bestimmung derjenigen Felder relevant, von denen aus der jeweils nächste Harkzug gemacht werden soll. Solche Felder sollen als *aktuelle Harkfelder* bezeichnet werden. Ein aktuelles Harkfeld zeichnet sich im dafür erstellten Algorithmus²⁰ dadurch aus, dass sämtliche Vorgänger im Verlaufe des Verfahrens bereits leer geharkt worden sind.²¹

Beispielsweise ergibt sich für die obige Nachfolgerfunktion s (vgl. Tab. 4) 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. 5).

Tabelle 5: 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 ist das jeweilige Hub). Entsprechend können sich die jeweils anschließenden Transportprozesse in ihren Transportaufwänden unterscheiden, ebenso die unproduktiven Wege. An diesem Beispiel

¹⁸ Man beachte, dass hier zunächst der reine Harkaufwand ohne Berücksichtigung von unproduktiven Wegen gemeint ist.

¹⁹ Hierzu siehe Algorithmus 5 im Anhang B

²⁰ Hierzu siehe Algorithmus 5 im Anhang B.

²¹ Diese 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. Allerdings 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 Ergänzung der Harkaufwandberechnung durch Einbeziehung dieser Wegaufwände wird auf den folgenden Abschn. 2.2.3 verwiesen.

wird deutlich, dass bei festgelegter Berechnungsgrundlage jeder vorgegebenen Nachfolgerfunktion s der kumulierte Harkaufwand $HA_{\Sigma}(s)$ zugewiesen werden kann.

In folgender Tabelle sind einige verschiedene Nachfolgerfunktionen aufgelistet. Es handelt sich dabei allerdings um solche „Spezialfälle“, bei denen stets *genau ein* Laubhaufen erzeugt wird.

Tabelle 6: Ausgewählte Nachfolgerfunktionen mit zugehörigem Harkaufwand

a	1	2	3	4	5	6	7	8	9	10	11	12	$HA_{\Sigma}(s_i)$
$s_1(a)$	2	6	3	8	9	10	11	12	10	11	11	11	47
$s_2(a)$	2	3	4	8	9	5	6	7	10	11	12	12	124
$s_3(a)$	2	3	4	4	1	2	3	4	5	6	7	8	76
$s_4(a)$	2	3	4	4	6	7	8	4	10	11	12	8	76
$s_5(a)$	2	6	3	8	6	7	7	7	5	11	7	8	46
$s_6(a)$	5	6	7	8	6	10	11	7	10	10	10	11	46
$s_7(a)$	5	6	7	8	6	7	8	8	5	6	7	8	59
$s_8(a)$	2	3	4	4	6	7	8	4	5	6	7	8	76

Es dürfte unverkennbar sein, dass die Anzahl der verschiedenen Nachfolgerfunktionen, insbesondere bei freier Wahl der Laubhaufenanzahl, 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 Harkstrategien zulässige Nachfolgerfunktionen zu generieren und ausgehend von einer zulässigen Nachfolgerfunktion bessere Nachfolgerfunktionen in der „Nachbarschaft“ zu finden, um damit das Laubharkproblem schrittweise zu lösen.

2.2.3 Berechnung von unproduktiven Wegezeiten

Es ist offensichtlich, dass durch die Nachfolgerfunktion s zwar einzelne Harkschritte und auch die grundsätzliche „Harkstruktur“ (Anzahl und Lage der Haufenfelder mit kumulierter Laubmenge) festgelegt werden, aber der gesamte Harkvorgang dennoch nicht eindeutig beschrieben wird, weil die Reihenfolge der einzelnen Harkschritte dadurch noch nicht festgelegt ist. Die folgende Darstellung der Nachfolgerfunktion s (vgl. Tab. 7) lässt offen, in welchem Feld der Harkprozess beginnen soll, wie nach einem Harkschritt $a \rightarrow s(a)$ fortgesetzt werden soll, insbesondere dann, wenn $s(a)$ noch Vorgängerfelder hat, die nicht leer geharkt sind, oder $s(a)$ ein Haufenfeld darstellt.

Tabelle 7: Beispiel einer Nachfolgerfunktion

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

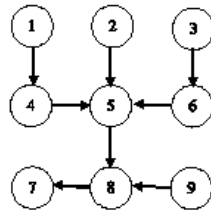


Abbildung 12: Digraph \vec{G}_s zur Nachfolgerfunktion s

Unter Beachtung des effizienten Harkprinzips, dass nur von Feldern geharkt werden sollte, deren Vorgängerfelder bereits leer geharkt sind, wird folgerichtig mit einer Harkquelle begonnen, und bei jeder zwangsläufigen Unterbrechung des produktiven Harkprozesses wird wieder bei einer noch nicht einbezogenen Harkquelle angesetzt. Für das obige Beispiel ergeben sich anhand der vier vorhandenen Harkquellen²² $4! = 24$ Möglichkeiten für effiziente Harkreihenfolgen.²³ In der folgenden Darstellung werden exemplarisch nur die vom Startknoten 1 ausgehenden Harkreihenfolgen aufgeführt; dabei werden produktive Harkschritte durch \rightarrow und unproduktive Wege durch \Rightarrow gekennzeichnet:

- Nr. 1: $1 \rightarrow 4 \rightarrow 5 \Rightarrow 2 \rightarrow 5 \Rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \Rightarrow 9 \rightarrow 8 \rightarrow 7$
 Nr. 2: $1 \rightarrow 4 \rightarrow 5 \Rightarrow 2 \rightarrow 5 \Rightarrow 9 \rightarrow 8 \Rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 7$
 Nr. 3: $1 \rightarrow 4 \rightarrow 5 \Rightarrow 3 \rightarrow 6 \rightarrow 5 \Rightarrow 2 \rightarrow 5 \rightarrow 8 \Rightarrow 9 \rightarrow 8 \rightarrow 7$
 Nr. 4: $1 \rightarrow 4 \rightarrow 5 \Rightarrow 3 \rightarrow 6 \rightarrow 5 \Rightarrow 9 \rightarrow 8 \Rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 7$
 Nr. 5: $1 \rightarrow 4 \rightarrow 5 \Rightarrow 9 \rightarrow 8 \Rightarrow 2 \rightarrow 5 \Rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 7$
 Nr. 6: $1 \rightarrow 4 \rightarrow 5 \Rightarrow 9 \rightarrow 8 \Rightarrow 3 \rightarrow 6 \rightarrow 5 \Rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 7$

Wird eine Entfernungsmatrix $D = (d_{ij})_{i,j=1,\dots,9}$ gemäß Manhattan-Metrik zu Grunde gelegt, ergeben sich für die obigen sechs Reihenfolgemöglichkeiten unterschiedliche Gesamtlängen für unproduktive Wege, wobei in diesem Beispiel der Einfachheit halber für die Kantenbewertungen $c_{ij} \equiv 1$ angenommen wird:

- Nr. 1: $d_{52} + d_{53} + d_{89} = 4$
 Nr. 2: $d_{52} + d_{59} + d_{83} = 6$
 Nr. 3: $d_{53} + d_{52} + d_{89} = 4$
 Nr. 4: $d_{53} + d_{59} + d_{82} = 6$
 Nr. 5: $d_{59} + d_{82} + d_{53} = 6$
 Nr. 6: $d_{59} + d_{83} + d_{52} = 6$

Somit ist die Minimierung des Aufwandes für unproduktive Wege bei einem durch eine Nachfolgerfunktion s beschriebenen Harkprozess auf die Bestimmung einer optimalen Harkquellenreihenfolge zurückgeführt. Es hat sich gezeigt [4], dass bei der Bestimmung

²² Die Menge der Harkquellen ist $Q(s) = \{1, 2, 3, 9\}$; vgl. Abb. 12.

²³ Man beachte, dass alle diese Harkreihenfolgen denselben Harkaufwand $HA_{\Sigma}(s)$ hervorrufen.

einer „guten“ Harkquellenreihenfolge das Greedy-Verfahren „Nächster Nachfolger“ effiziente Lösungen anbietet. Hiernach wird ausgehend von einem beliebigen Startknoten $a \in V$ die nächstgelegene Harkquelle k bestimmt:²⁴

$$\text{Wähle } k \in Q(s) \text{ mit } d_{ak} = \min\{d_{aj} \mid j \in Q(s)\}. \quad (2.13)$$

Nach diesem Prinzip wird jede weitere Harkquelle bestimmt, sobald der produktive Harkprozess unterbrochen werden muss, weil der jeweilige aktuelle Knoten a_i ein Hub ist oder Vorgängerknoten besitzt, die noch nicht leer geharkt sind. Dabei wird jede Harkquelle genau einmal in Anspruch genommen.²⁵ Entsprechend der ermittelten Reihenfolge der Harkquellen q_1, q_2, \dots, q_Q werden die Entfernungen zwischen dem jeweils aktuellen Knoten a_{i-1} und der jeweils neuen Harkquelle q_i kumuliert, beginnend mit einem Startknoten a_0 .²⁶

$$W(s) = \sum_{i=1}^Q d_{a_{i-1}, q_i}. \quad (2.14)$$

Die Gesamtlänge $W(s)$ aller unproduktiven Wege $a_{i-1} \Rightarrow q_i$ für $i = 1, \dots, Q$ werden noch mit einem *Wegeaufwandsfaktor* α_W gewichtet:

$$WA_\Sigma(s) = \alpha_W \cdot W(s) = \alpha_W \cdot \sum_{i=1}^Q d_{a_{i-1}, q_i}. \quad (2.15)$$

Der Wegeaufwandsfaktor α_W hat die Dimension „Zeitkosteneinheit pro Längeneinheit“ $[ZE/LE]$, wenn die Entfernungen d_{ij} bzw. die Kantenbewertungen c_{ij} nach Längeneinheiten $[LE]$ gemessen werden. Als Verallgemeinerung eines konstanten Faktors α_W können auch von den jeweiligen Entfernungen d_{ij} abhängige Wegeaufwandsfaktoren $\alpha_W(i, j)$ eingeführt werden; dann ändert sich die Formel (2.15) in

$$WA_\Sigma(s) = \sum_{i=1}^Q \alpha_W(a_{i-1}, q_i) \cdot d_{a_{i-1}, q_i}. \quad (2.16)$$

Um die Berechnung des unproduktiven Wegeaufwandes einerseits und gleichzeitig auch des produktiven Harkaufwandes gemäß (2.12) andererseits aus der tabellarischen Darstellung einer Nachfolgerfunktion s unmittelbar „ablesbar“ zu machen, wird die Nachfolgerfunktion s mittels einer Permutation $\pi : V \leftrightarrow V$ ausgerichtet, indem der Wert $\pi(a)$ angibt, in welcher Spalte der neu ausgerichteten Tabelle das Paar $\begin{vmatrix} a \\ s(a) \end{vmatrix}$ positioniert wird. Der durch π hervorgerufene Spaltentausch beginnt mit dem Knoten a_1 , bei dem der Harkprozess beginnen soll, d.h. $\pi(a_1) = 1$. Danach wird vom jeweils aktuellen Knoten a_i gemäß s zum Knoten $s(a_i)$ geharkt, wenn a_i weder Haufenfeld ist

²⁴ Falls der Startknoten a selbst eine Harkquelle ist, gilt $k = a$ wegen $d_{aa} = 0$.

²⁵ Die Mächtigkeit der durch s induzierten Harkquellenmenge $Q(s)$ werde mit $Q := |Q(s)|$ bezeichnet.

²⁶ Hierzu siehe Algorithmus 6 im Anhang B.

2 Das Laubharkproblem als kombinatorisches Optimierungsproblem

noch nichtleere Vorgängerknoten besitzt; dabei wird $\pi(a_i) = i$ gesetzt und $a_{i+1} = s(a_i)$ als nächster aktuelle Knoten betrachtet. Umgekehrt wird im Falle einer Harkunterbrechung eine jeweils neue Harkquelle $q \in Q(s)$ als nächster aktueller Knoten gewählt, d.h. $a_{i+1} = q$.

In der folgenden Tabelle ist die Nachfolgerfunktion s aus Tab. 7 an der obigen Reihenfolgemöglichkeit Nr. 3 ausgerichtet.

Tabelle 8: Beispiel einer ausgerichteten Nachfolgerfunktion

$\pi(a)$	1	2	3	4	5	6	7	8	9
a	1	4	3	6	2	5	9	8	7
$s(a)$	4	5	6	5	5	8	8	7	7

Anhand dieser Tabelle lässt sich die Harkaufwandsberechnung sowohl der produktiven Harkschritte (\downarrow) als auch der unproduktiven Wege (\nearrow) einfach ablesen, wobei durch \nearrow und \downarrow bewertungsfreie Reihenfolgefortsetzungen angedeutet werden²⁷:

Tabelle 9: Auswertung einer ausgerichteten Nachfolgerfunktion

a	1	4	3	6	2	5	9	8	7
	$\downarrow \nearrow$	$\downarrow \nearrow$	$\downarrow \nearrow$	$\downarrow \nearrow$	$\downarrow \nearrow$	$\downarrow \nearrow$	$\downarrow \nearrow$	$\downarrow \nearrow$	\downarrow
$s(a)$	4	5	6	5	5	8	8	7	7

Entlang dieser Zickzack-Reihenfolge werden bei jedem schwarzen Pfeil \downarrow die produktiven Harkaufwände $HA(a, s(a))$ mit $a \neq s(a)$ und bei jedem rotem Pfeil \nearrow die unproduktiven Wegeaufwände $\alpha_W \cdot d_{s(\pi^{-1}(i)), \pi^{-1}(i+1)}, i = 1, \dots, n-1$, kumuliert. An obigem Beispiel ergibt sich die folgende Rechnung²⁸:

$$HA(1, 4) + HA(4, 5) + \alpha_W(5, 3) \cdot d_{5,3} + HA(3, 6) + HA(6, 5) + \alpha_W(5, 2) \cdot d_{5,2} \\ + HA(2, 5) + HA(5, 8) + \alpha_W(8, 9) \cdot d_{8,9} + HA(9, 8) + HA(8, 7).$$

2.2.4 Berechnung des Transportaufwandes

Für das Aufwandsmodell des gesamten Laubharkproblems wird nun noch der Transportprozess einbezogen. Hierbei spielen zurückzulegende Transportwege zwischen den Hubs und dem Kompostfeld K die entscheidende Rolle.²⁹

²⁷ Zur Ausrichtung einer Nachfolgerfunktion dient Algorithmus 7 im Anhang B.

²⁸ Zur simultanen Berechnung von Hark- und Wegeaufwänden anhand einer ausgerichteten Nachfolgerfunktion dient Algorithmus 8 im Anhang B.

²⁹ Zur Einbeziehung des Kompostfeldes in das Graphenmodell gibt es grundsätzlich zwei Möglichkeiten: Das Kompostfeld ist ein bestimmter Knoten des Graphen ($K \in V$) oder ein zusätzlicher Knoten ($V^* = V \cup \{K\}$), allerdings jeweils mit hinreichend großer Aufnahmekapazität ($\overline{M}(K) = \infty$). Im Folgenden wird o.B.d.A. die erste Möglichkeit verfolgt.

Ausgehend von einer Nachfolgerfunktion s sei die Hubmenge $Hub(s)$ ermittelt³⁰ und ein Kompostfeld $K \in V$ vorgegeben. Die Entfernungen zwischen allen Hubs $h \in Hub(s)$ untereinander und zum Kompostfeld K liegen in einer Teilmatrix $D^* = (d_{hk}^*)_{h,k \in Hub(s) \cup \{K\}}$ der Entfernungsmatrix $D = (d_{ij})_{i,j \in V}$ vor. Alle Entfernungen haben die Dimension [LE] (Längeneinheit).

Der Transportaufwand ließe sich dann proportional zu den zurückgelegten Weglängen bestimmen. Bei einer Pendeltour zwischen dem Kompostfeld K und einem Hub h ergibt sich im einfachsten Fall der Transportaufwand

$$TA(h) = 2 \cdot d_{h,K}^* \cdot \alpha_T, \quad (2.17)$$

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

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

$$TA(h) = 2 \cdot d_{h,K}^* \cdot \alpha_T + \gamma \cdot M_s^*(h) + \sigma. \quad (2.18)$$

Hierbei geht der Parameter γ in den variablen Aufwandsanteil für das jeweilige Auf- und Abladen der Laubmenge $M^*(h)$ ein und hat die Dimension [ZE/ME], während die Konstante σ für den fixen Aufwandsanteil steht und die Dimension [ZE] hat.³¹

Bislang wird unterstellt, dass die gesamte Laubmenge $M^*(h)$, die vom Hub h abgeholt wird, auch komplett vom Transportmittel (Laubwagen oder Schubkarre) aufgenommen werden kann. Allerdings kann es vorkommen, dass die vom Hub h abzuholende Laubmenge das *maximale Ladevolumen* \bar{T} der Schubkarre überschreitet. In diesem Falle sind mehrere Pendeltouren nötig, um das Hub h vom Laubhaufen zu befreien. Entsprechend ändert sich der Transportaufwand für die Laubentsorgung des Haufenfeldes h :

$$TA(h) = \left\lceil \frac{M_s^*(h)}{\bar{T}} \right\rceil \cdot 2 \cdot d_{h,K}^* \cdot \alpha_T \quad (2.19)$$

³⁰ Vgl. Algorithmus 1.

³¹ Eine Abhängigkeit der Parameter γ und σ von den Hubfeldern h als mögliche Verallgemeinerung, also $\gamma(h)$ und $\sigma(h)$, wird hier nicht weiter betrachtet. Es sei allerdings darauf hingewiesen, dass bei der späteren Aufsummierung der einzelnen Transportaufwände $TA(s)$ über alle Hubs (siehe (2.21)) ein konstanter Parameter γ keinen maßgeblichen Einfluss auf die Lösungsgüte einer Harkstrategie (bzw. der dadurch erzeugten Nachfolgerfunktion s) ausübt, wohl aber ein konstanter Parameter σ , da die Anzahl der durch s induzierten Hubs hierbei eine Rolle spielt.

bzw.

$$TA(h) = \left\lceil \frac{M_s^*(h)}{\bar{T}} \right\rceil \cdot 2 \cdot d_{h,K}^* \cdot \alpha_T + \gamma \cdot M_s^*(h) + \left\lceil \frac{M_s^*(h)}{\bar{T}} \right\rceil \cdot \sigma. \quad (2.20)$$

Liegen mehrere Laubhaufen zum Abholen bereit, ergibt sich der gesamte Transportaufwand als die Summe aller Pendeltouren zu allen Hubs (bzgl. Nachfolgerfunktion s):

$$TA_\Sigma(s) = \sum_{h \in Hub(s)} TA(h). \quad (2.21)$$

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 Hubs in ihrer Laubmengensumme das maximale Ladevolumen \bar{T} nicht überschreiten. Im Folgenden wird also davon ausgegangen, dass nur Sammeltouren $K \rightarrow h_1 \rightarrow h_2 \rightarrow \dots \rightarrow h_P \rightarrow K$ betrachtet werden, für die gilt:

$$\sum_{p=1, \dots, P-1} M_s^{**}(h_p) < \bar{T}, \quad (2.22)$$

wobei mit $M_s^{**}(h_p)$ die nach entsprechend vielen Pendeltouren verbleibende Laubrestmenge beim Hub h_p gemeint ist. Hiermit wird sichergestellt, dass bei Weiterfahrt zum nächsten Hub 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 $M_s^{**}(h_P)$ auf der Schubkarre Platz findet. Dann ergibt sich der Transportaufwand für diese Sammeltour im einfachen Fall, d.h. in Anlehnung an (2.17), zu

$$TA(h_1, \dots, h_P) = \alpha_T \cdot [d_{K,h_1}^* + d_{h_P,K}^* + \sum_{p=2, \dots, P} d_{h_{p-1},h_p}^*]. \quad (2.23)$$

Auf weitere Verfeinerungen soll hier verzichtet werden; hierzu siehe [5], S. 79.

Zur Ermittlung von effizienten Transportabfolgen, insbesondere bei den Sammeltouren, empfiehlt sich der Einsatz von Lösungsverfahren für Tourenplanungsprobleme. Das im Anhang vorgestellte Verfahren (vgl. Algorithmus 9 im Anhang B) benutzt vereinfacht das „Nächster Nachfolger“-Prinzip.

2.3 Das formale Laubharkmodell

Die in den obigen Abschnitten dieses Kapitels ausführlich hergeleitete mathematische Modellierung eines Laubharkproblems soll hier auf seinen formalen Kern zusammengefasst werden.

Gegeben sei ein kanten- und knotenbewerteter Graph $G = [V, E, c, M]$ mit der Knotenmenge V , der Kantenmenge E , der Kantenbewertungsfunktion $c : E \rightarrow \mathbb{R}_{\geq}$ und der Knotenbewertungsfunktion $M : V \rightarrow \mathbb{R}_{\geq}$.

Eine Abbildung $s : V \rightarrow V$ heißt *Nachfolgerfunktion* von G , wenn s die *Nachbarschaftsbedingung* erfüllt (d.h. für alle $a \in V$ gilt: $a \neq s(a) \Rightarrow [a, s(a)] \in E$; vgl. (2.9)), und wenn zudem der durch s induzierte Digraph $\vec{G}_s = \langle V, \vec{E}_s \rangle$ mit $\vec{E}_s = \{\langle a, s(a) \rangle \mid a \in V \wedge a \neq s(a)\}$ zyklensfrei ist. Die Menge aller Senken in \vec{G}_s werde mit $Hub(G, s)$ bezeichnet.

Zu jedem $a \in V$ ist bzgl. \vec{G}_s die *Erreichbarkeitsmenge* $R(a)$ definiert, d.h. die Menge aller Knoten $i \in V$, von denen aus der Knoten a über eine Pfeilfolge in \vec{G}_s erreichbar ist; dabei werde $a \in R(a)$ vereinbart. Die Erreichbarkeitsmengen der Senken in \vec{G}_s werden auch *Cluster* genannt und mit $C(h)$ für $h \in Hub(G, s)$ bezeichnet. Zudem wird die Abbildung $M_s^* : V \rightarrow \mathbb{R}_{\geq}$ durch $M_s^*(a) := \sum_{k \in R(a)} M(k)$ definiert; sie wird die durch s induzierte *Anhäufungsfunktion* genannt.

Die Menge aller Nachfolgerfunktionen von G werde mit $\Lambda(G)$ bezeichnet. Durch eine *Kostenfunktion* $K : \Lambda(G) \rightarrow \mathbb{R}_{\geq}$ werden jeder Nachfolgerfunktion s von G nichtnegative Kosten zugewiesen.³² Dann wird das Optimierungsproblem in der Form

$$\min_{s \in \Lambda(G)} K(s) \quad (2.24)$$

ein *Laubharkproblem* genannt. Dabei wird eine kostenminimale Nachfolgerfunktion s von G gesucht.

Mit der Abbildung $\overline{M} : V \rightarrow \mathbb{R}_{\geq}$ werde eine *Kapazitätsfunktion* eingeführt.³³ Eine Nachfolgerfunktion s heißt *zulässig* bzgl. \overline{M} , wenn für alle Knoten $a \in V$ gilt: $M_s^*(a) \leq \overline{M}(a)$.³⁴ Dann wird das Optimierungsproblem in der Form

$$\begin{aligned} & \min_{s \in \Lambda(G)} K(s) \\ & \text{u.d.N.} \\ & M_s^*(a) \leq \overline{M}(a) \text{ für alle } a \in V \end{aligned} \quad (2.25)$$

³² Spezielle Kostenfunktionen ergeben sich durch die Aufsummierung der Aufwände für Harken, unproduktive Wege und Transporte. So könnte die Kostenfunktion folgende Form haben: $K(s) = HA_{\Sigma}(s) + WA_{\Sigma}(s) + TA_{\Sigma}(s)$ gemäß (2.8), (2.15) und (2.21).

³³ Im einfachsten Fall gilt $\overline{M}(a) = \overline{M} \in \mathbb{R}_{>}$ für alle $a \in V$.

³⁴ Vgl. (2.11).

2 Das Laubharkproblem als kombinatorisches Optimierungsproblem

ein *kapazitiertes Laubharkproblem* genannt.³⁵ Dabei wird eine kostenminimale zulässige Nachfolgerfunktion s von G gesucht.

³⁵ Die Nebenbedingung kann im Falle einer konstanten Kapazitätsfunktion (d.h. $\overline{M}(a) = \overline{M} \in \mathbb{R}_{>}$ für alle $a \in V$) auch allein nur für die Senken $h \in \text{Hub}(s)$ gefordert werden, ohne dass sich dadurch eine wesentliche Änderung ergibt.

3 Lösungsstrategien für das Laubharkproblem

Im Folgenden wird davon ausgegangen, dass ein kapazitiertes Laubharkproblem gemäß (2.25) zu Grunde liegt, für das eine effiziente Lösung in Form einer Nachfolgerfunktion s eines gegebenen knoten- und kantenbewerteten Graphen $G = [V, E, c, M]$ gesucht wird. Dabei setze sich die zu minimierende Zielfunktion $K(s)$ aus den additiven Komponenten $HA_{\Sigma}(s)$ gemäß (2.8), $WA_{\Sigma}(s)$ gemäß (2.15) und $TA_{\Sigma}(s)$ gemäß (2.21) zusammen:

$$K(s) = HA_{\Sigma}(s) + WA_{\Sigma}(s) + TA_{\Sigma}(s) \quad (3.1)$$

Zu einer ermittelten Nachfolgerfunktion s kann mittels der im Anhang B vorgestellten Algorithmen sowohl die Zulässigkeit geprüft als auch der zugehörige Zielfunktionswert errechnet werden (hierzu siehe auch Kapitel ??).

Die Bestimmung einer effizienten Lösung³⁶ erfolgt in der Regel in zwei Stufen. Zunächst wird via *Eröffnungs-* oder *Konstruktionsverfahren* eine zulässige Anfangslösung ermittelt. Anschließend wird diese Startlösung mittels *Verbesserungsverfahren* schrittweise verbessert. Darüber hinaus lassen sich die Harkstrategien im Wesentlichen anhand der Reihenfolge unterscheiden, nach der zum einen die Laubhaufen (Hubs) und zum anderen die diesen Laubhaufen jeweils zugewiesenen Felder (Cluster) bestimmt werden: *Hub first – Cluster second* oder *Cluster first – Hub second*. Darüber hinaus gibt es auch Verfahren, bei denen die Clusterbildung und die Hubbestimmung simultan vollzogen werden (z.B. bei Zickzack-Strategien; vgl. Abschn. 3.2.1).

3.1 Hubbestimmung in einem Cluster

Zunächst wird der Begriff des Clusters formalisiert und die Bestimmung eines effizienten Hubs in einem Cluster sowie die effiziente Ausrichtung aller „Nicht-Hubs“ auf das Hub beschrieben.

³⁶ Da es sich bei den Verfahren um heuristische Methoden handelt, wird der Begriff „optimal“ (bzw. „minimal“) tunlichst vermieden.

Gegeben sei ein knoten- und kantenbewerteter Graph $G = [V, E, M, c]$ mit Entfernungsmatrix $D = (d_{ij})$. Eine nichtleere Teilmenge $C \subseteq V$ heißt ein *Cluster* von G , wenn der durch C aufgespannte Teilgraph $G|_C = [C, E|_C]$ zusammenhängend ist.³⁷ Eine Zerlegung der Knotenmenge V in Cluster C_1, \dots, C_k heißt auch eine *Clustering* von G . Dabei soll jedes Cluster einer Clustering durch genau einen Knoten repräsentiert werden.

3.1.1 Hubzentrierung in einem Cluster

Zu einem gegebenen Cluster C soll *genau ein* Knoten $h \in C$ als Hub zugewiesen werden. Dadurch wird jedes Cluster eindeutig durch *sein* Hub identifiziert: $C = C(h)$. Hierfür bieten sich verschiedene Auswahlkriterien an:

Ein sehr einfaches und intuitives Auswahlkriterium besteht darin, dass ein Knoten mit maximaler Laubmenge als Hub bestimmt wird³⁸ (*Maximalmengen-Kriterium*):

$$\text{Bestimme } h \in C \text{ mit } M(h) = \max\{M(a) \mid a \in C\}. \quad (3.2)$$

Aufwändiger ist die Bestimmung eines Knotenmedians in $G|_C$. Zunächst wird jedem Knoten $a \in C$ der Wert $M_C^*(a) := \sum_{j \in C \setminus \{a\}} d_{aj}^C \cdot M(j)$ zugewiesen, wobei d_{aj}^C die Länge eines kürzesten Weges zwischen den Knoten a und j in $G|_C$ ist.³⁹ Dieser Wert gibt den kumulierten Harkaufwand von allen Knoten in C zum jeweiligen Knoten a an.⁴⁰ Das *Median-Kriterium*⁴¹ lautet demnach:

$$\text{Bestimme } h \in C \text{ mit } M_C^*(h) = \min\{M_C^*(a) \mid a \in C\}. \quad (3.3)$$

Ebenso wie das obige Maximalmengen-Kriterium zielt erst recht das Median-Kriterium auf eine Reduzierung des Harkaufwandes. Aufgrund des hohen Rechenaufwandes bietet sich die Knotenmedianbestimmung allerdings eher als ein Sekundärkriterium an. Das *Kompostfeld-Kriterium* zielt hingegen auf eine Verringerung des Transportaufwandes ab, indem das Harken in Richtung des Kompostfeldes K ausgerichtet wird:

$$\text{Bestimme } h \in C \text{ mit } d_{hK} = \min\{d_{aK} \mid a \in C\}. \quad (3.4)$$

Die drei vorgestellten Auswahlkriterien müssen nicht zu einer eindeutigen Lösung führen, selbst wenn sie – in welcher Reihenfolge auch immer – als Primär-, Sekundär- oder

³⁷ Ein aufgespannter Teilgraph, auch *Untergraph* genannt, ist „kantenmaximal“, d.h. $i, j \in C \wedge [i, j] \in E \Rightarrow [i, j] \in E|_C$.

³⁸ Vgl. [6], S. 96.

³⁹ Man beachte: $d_{aj}^C \geq d_{aj}$.

⁴⁰ Man beachte, dass die Berechnung des produktiven Harkaufwandes hierdurch in gewisser Weise bereits vorweggenommen wird.

⁴¹ Zur Bestimmung von Knotenmedianen in Graphen siehe auch [2], S. 41ff.

Tertiärkriterium eingesetzt werden. Daher bietet es sich an, die Eindeutigkeit der Lösung durch ein abschließendes *Indexkriterium* zu erzwingen:

$$\text{Bestimme } h \in C \text{ mit } h = \begin{cases} \min\{a \mid a \in C\} \\ \max\{a \mid a \in C\} \end{cases} \quad (\text{wahlweise}). \quad (3.5)$$

3.1.2 Hubausrichtung in einem Cluster

Ist das Hub h in einem Cluster C bestimmt, werden alle anderen Knoten des Clusters auf das Hub ausgerichtet. Hierzu gibt es grundsätzlich zwei Möglichkeiten:

- (1) Es wird beim Hub h begonnen und $s(h) = h$ gesetzt. Dann werden sämtliche Nachbarn $b \in C$ des Hubs h ermittelt und $s(b) = h$ gesetzt. Diese Nachbarn werden in einem Schlangenspeicher L aufgenommen. Nach dem FIFO-Prinzip werden der Reihe nach die Knoten aus dem Speicher L entnommen; zu jedem entnommenen Knoten $a \in L$ werden sukzessiv dessen Nachbarn $b \in C$, die noch keine Zuweisung $s(b)$ bekommen haben, ermittelt und dem Stapelspeicher L zugeführt; dabei wird $s(b) = a$ gesetzt. Die Zuweisungsprozedur endet, wenn alle Knoten $a \in C$ auf diese Weise eine Zuweisung $s(a)$ erhalten haben. Hierzu siehe Algorithmus 14 im Anhang B.
- (2) Zu jedem Knoten $a \in C$ werden die Entfernungen d_{ah}^C zum Hub h innerhalb des von C aufgespannten Teilgraphen $G|_C$ ermittelt; dabei handelt es sich nicht notwendigerweise um die Entfernung d_{ah} in V . Daraus wird die maximale Entfernung d_{max} ermittelt. Beginnend bei den Knoten $a \in C$ mit maximaler Entfernung zum Hub (d.h. $d_{ah}^C = d_{max}$) wird jeweils ein Nachbar $b \in C$ mit $d_{bh}^C = d_{max} - 1$ ermittelt und $s(a) = b$ gesetzt. Der Wert d_{max} wird nun schrittweise um 1 verringert, wobei in jedem Schritt die obige Prozedur wiederholt wird, bis alle Knoten $a \in C$ auf diese Weise eine Zuweisung $s(a)$ erhalten haben. Hierzu siehe Algorithmus 14* im Anhang B.

Auf diese Weise wird eine (auf C eingeschränkte) Nachfolgerfunktion s von G konstruiert, deren Zulässigkeit bzgl. einer Kapazitätsfunktion \overline{M} im Wesentlichen von einer geeigneten Bestimmung des Clusters C abhängt.

3.1.3 Sonderbehandlung von Leerfeldern

Bei der Bestimmung von Clustern und der Berechnung von Hark- und unproduktiven Wegeaufwänden bzgl. einer Nachfolgerfunktion s kann es sinnvoll sein, dass *Leerfelder* (d.h. $M(a) = 0$) einer speziellen Behandlung unterzogen werden. Einerseits ist es nicht opportun, leere Felder als Harkquellen zu betrachten, auch wenn der Harkaufwand von einem leeren Feld a zu einem Nachbarfeld b per Rechenvorschrift keinen Harkaufwand

hervorrufen, wohl aber unproduktiven Wegeaufwand, der ggf. nicht explizit in Rechnung gestellt wird. Andererseits führt die Zugehörigkeit eines Leerfeldes zu einem Cluster bei den im Folgenden beschriebenen Laubharkstrategien dazu, dass über dieses Leerfeld nur innerhalb eines Clusters hinweggeharkt werden darf, wodurch sowohl die Clusterbildung als auch die Bestimmung eines (möglichst realitätsnahen) Harkplans unnötig einschränkt werden kann.

Daher wird vorgeschlagen, die Menge der Leerfelder (ggf. wahlweise) als ein spezielles Cluster $C_0 := \{a \in V \mid M(a) = 0\}$ aufzufassen und die Sonderstellung der Felder dieses Clusters bei der Implementierung der Harkstrategien passend zu berücksichtigen. Diese Möglichkeit einer Sonderbehandlung wird im Folgenden allerdings nicht weiter explizit betrachtet.

3.2 Konstruktionsverfahren

Die im Folgenden vorgestellten Konstruktionsverfahren orientieren sich an Harkstrategien beim intuitiv praktizierten Harken in einem Garten.

3.2.1 Zickzack-Strategien

Vom nordwestlichen Feld (o.B.d.A. Feld 1) wird solange in waagerichter Richtung geharkt, beginnend von Westen nach Osten bzw. von links nach rechts, bis die jeweilige Gartengrenze erreicht ist oder die kumulierte Harklaubmenge die vorgegebenen Kapazitäten \overline{M} der Felder überschreitet. Im ersten Fall wird senkrecht in das untere Feld und von dort in die umgekehrte waagerechte Richtung weiter geharkt (falls möglich), also abwechselnd von links nach rechts bzw. von rechts nach links. Im zweiten Fall wird das letzte Feld, welches die kumulierte Laubmenge noch aufnehmen kann, zum Haufenfeld (Hub) erklärt und es wird beim nächsten harkbaren Feld in jeweilige Harkrichtung mit dem Laubharken wieder neu begonnen. Falls man bei diesem Zickzack-Kurs auf ein Ausschlussfeld stößt, wird auch hier das vorige Feld zum Hub erklärt und das nächste zulässige Feld in Zickzack-Richtung gesucht, um dort mit dem Harken fortzusetzen (vgl. Abb. 13).

→	→	→	→	→	→	→	→	→	→	h ₁	→	↓
↓	←	←		h ₂	←	←	←	←	←	←	←	←
→	→	→	h ₃			→	→	→	→	→	→	↓
h ₅	←	←	←	←	←	←	←	←	←	h ₄	←	←
→	→	↓
	h _k	←	←

Abbildung 13: Zickzack-Strategie

Aufgrund dieser Harkvorschrift ist die Abbildung $s : V \rightarrow V$ gemäß

$$s(a) := \begin{cases} a, & \text{falls } a \text{ ein Hub ist} \\ a + 1, & \text{falls von } a \text{ nach rechts geharkt wird} \\ a - 1, & \text{falls von } a \text{ nach links geharkt wird} \\ a + n, & \text{falls von } a \text{ nach unten geharkt wird} \end{cases} \quad (3.6)$$

eine zulässige Nachfolgerfunktion von G . Die Anwendung dieser *einfachen Zickzack-Strategie* wird im Anhang C.1.1 an einem Beispiel demonstriert. Eine algorithmische Beschreibung dieser einfachen Zickzack-Strategie findet sich als Algorithmus 10 im Anhang B.

Verbindet man die Knoten entlang dieser Harkvorschrift durch Pfeile, entstehen Pfade⁴², die in ihrer Gesamtheit jeden Knoten $a \in V$ genau einmal enthalten und somit einen speziellen Digraphen \vec{G} aufspannen, wobei jeder Pfad einen gerichteten Baum darstellt, dessen einziges Blatt⁴³ ein (vorläufiges) Haufenfeld des Harkprozesses darstellt. Jeder Pfad liefert auf kanonische Weise ein Cluster, in welchem durch Hubzentrierung ein neues Hub bestimmt und durch Ausrichtung auf dieses neue Hub die Nachfolgerfunktion geändert werden kann (hierzu siehe auch das Anwendungsbeispiel im Anhang C.1.1).

Diese Zickzack-Strategie lässt sich dahingehend verfeinern, indem sie auf Teilmatrizen angewendet wird, um beispielsweise lange waagerechte „Harkpfade“ zu vermeiden. Hierfür wird die Indexfolge $j = 1, \dots, n$ in mehrere (wenn möglich äquidistante) Teilfolgen unterteilt: $1, \dots, n_1, n_1 + 1, \dots, n_2, \dots, n_{k-1}, n_{k-1} + 1, \dots, n_k, \dots, n$. Die Anwendung dieser *unterteilten Zickzack-Strategie* wird im Anhang C.1.2 an einem Beispiel demonstriert.

⁴² Ein Pfad ist eine Pfeilfolge, die jeden Knoten $a \in V$ höchstens einmal enthält.

⁴³ Die Senken eines gerichteten Baumes werden auch *Blätter* genannt.

3.2.2 Clusterorientiertes Konstruktionsverfahren

Die hier beschriebene Clusterbildung beginnt o.B.d.A. in der Nordwestecke des Gartens; diese entspricht dem Knoten mit dem kleinsten Index ($a = 1$). Es werden alle Nachbarknoten von a ermittelt und nach aufsteigenden Indizes einem Schlangenspeicher N zugeführt. Danach wird schrittweise ein Knoten $b \in N$ entnommen und geprüft, ob die Hinzunahme in das Cluster von a die Laubmengenbeschränkung \overline{M} überschreitet. Wenn nicht, wird der Knoten b ins Cluster von a übernommen und aus der Suchliste gestrichen; gleichzeitig werden alle Nachbarn von b , die noch keine Clusterzuweisung haben, der Kandidatenliste N nach obigem Prinzip zugeführt. Diese Aufnahmeprozedur wird solange wiederholt, bis die Liste N leer ist. Danach wird ein neuer Knoten a bestimmt (mit kleinstem Index) und ein weiteres Cluster gebildet.⁴⁴

Es bietet sich an, jedem einzelnen Cluster in einem zweiten Schritt ein Hub zuzuweisen (siehe Abschn. 3.1.1) und die übrigen Knoten auf das Hub auszurichten (vgl. Abschn. 3.1.2).

3.2.3 Huborientiertes Konstruktionsverfahren

Beim folgenden Konstruktionsverfahren⁴⁵ beginnt jede Iteration ℓ mit der Suche nach einem (weiteren) Hub h_ℓ , indem ein Knoten mit der größten Laubmenge unter allen noch nicht einem Cluster zugeordneten Knoten von G bestimmt wird.⁴⁶ Ist ein Hub h_ℓ bestimmt, wird die kumulierte Laubharkmenge $M^*(h_\ell)$ mit der Laubmenge $M(h_\ell)$ initialisiert. Dann geht das Verfahren in die Nachbarsuche. Hierbei werden ausgehend vom jeweils aktuellen Knoten die Nachbarknoten gesucht, die unter bestimmten Bedingungen dem Cluster $C(h_\ell)$ zugeordnet werden. Wird also die Nachbarsuche mit einem Knoten a gestartet, werden zu diesem Knoten zuerst einmal alle Nachbarn, d.h. alle Knoten b mit $[a, b] \in E$, als potentielle Harkvorgänger bestimmt und auf folgende Eigenschaften geprüft:

1. Der Nachbarknoten b besitzt noch keine Zuweisung $s(b)$, ist also noch keinem Cluster zugeordnet.
2. Die Obergrenze \overline{M} wird nicht überschritten, wenn die Laubmenge $M(b)$ des Nachbarn b der aktuellen Harklaubmenge $M^*(h_\ell)$ hinzugefügt wird.
3. Die Laubmenge $M(b)$ dieses Nachbarn ist nicht größer als die Laubmenge $M(a)$ des aktuellen Knotens a , also $M(b) \leq M(a)$.

⁴⁴ Eine algorithmische Beschreibung dieser Clusterbildung ist im Anhang B als Algorithmus 11 zu finden. Zur Anwendung des Verfahrens an einem Beispiel siehe Anhang C.2.

⁴⁵ In enger Anlehnung an die in [6], S. 97ff, beschriebene „dynamische Harkstrategie“.

⁴⁶ Vgl. *Maximalmengen-Kriterium* (3.2)

Die erste Bedingung stellt sicher, dass jeder Knoten nur einem Cluster zugeordnet wird, indem b höchstens einen Harknachfolger $s(b)$ erhält (Eindeutigkeit der Harkfolge). Das zweite Prüfkriterium garantiert, dass die Harklaubmenge innerhalb eines Clusters die maximale Laubmenge \overline{M} nicht überschreitet (Zulässigkeit). 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 Harkvorgänger 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 den Knoten b zutreffen, wird es in die *Menge der potenziellen Harkvorgänger*, B_{pot} , aufgenommen. Welchen Nachbarn b nun das aktuelle Feld a als Harkvorgänger erhält, entscheidet wieder deren Laubmenge, indem nur derjenige potentielle Nachbar entweder mit der größten Laubmenge (vgl. Maximalmengen-Kriterium (3.2)) oder der kleinsten Laubmenge (*Minimalmengen-Kriterium*) schließlich durch die Setzung $s(b) = a$ in das Cluster $C(h_\ell)$ aufgenommen wird und erhält somit das betrachtete Feld a als Harknachfolger. Nachdem ein Knoten b aufgenommen und seine Laubmenge zur kumulierten Harklaubmenge hinzuaddiert wurde, wird nun erneut die Nachfolgersuche aufgerufen, diesmal mit dem soeben hinzugefügten Knoten b . Wird kein weiterer Nachfolger gefunden, finden systematische Rückwärtsschritte (*Backtracking*) statt. Dabei wird jeder Knoten wiederholt der Nachbarnsuche unterzogen. Dieser Zuordnungsprozess wird im Folgenden exemplarisch innerhalb eines einzigen Clusters nachvollzogen (vgl. Abb. 14). Dabei ist das aktuell betrachtete Feld jeweils schwarz umrandet.

Die Suche wird erst abgebrochen, wenn das Hub keine potenziellen Harkvorgänger mehr findet oder die maximale Laubharkmenge \overline{M} erreicht ist. Trifft eines zu, wird die Suchfunktion beendet und geprüft, ob noch Knoten existieren, die keinem Cluster zugeordnet sind. Unter diesen Knoten wird dann das nächste Hub gesucht und somit die nächste Iteration gestartet.⁴⁷

⁴⁷ Die algorithmische Beschreibung findet sich als Algorithmus 12 im Anhang B.

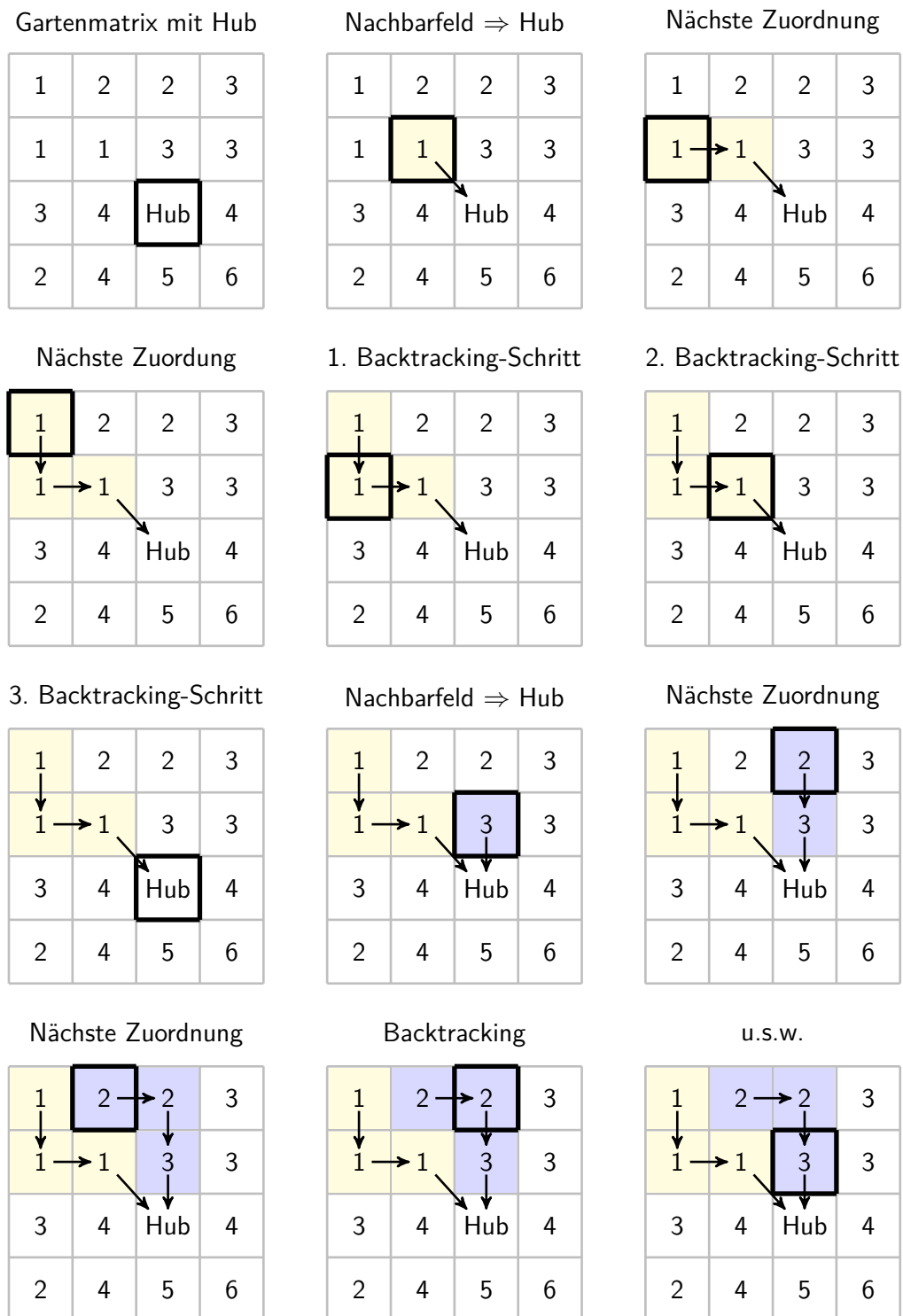


Abbildung 14: Zuordnungsprozess beim huborientierten Verfahren (TS) mit Minimalmengen-Kriterium (vgl. [6], S. 99)

Der mit diesem Konstruktionsverfahren ermittelte „Harkplan“ kann sich durch das Auswahlkriterium für den jeweiligen Harkvorgänger b (Maximal- oder Minimalmengen-Kriterium) enorm unterscheiden. Darüber hinaus sei angemerkt, dass die obige Beschreibung anhand der Bestimmungskriterien eine Tiefensuche (TS) induziert. Alternativ lässt sich auch eine Breitensuche (BS) implementieren⁴⁸, die ebenfalls zu stark abwei-

⁴⁸ Hierzu siehe Algorithmus 13 im Anhang B.

chenden Harkplänen führen kann. Die Unterschiedlichkeit der Harkpläne anhand der verschiedenen Verfahrensvarianten werden im Anhang C.3 demonstriert.

3.3 Verbesserungsverfahren

Die zwei vorgestellten Konstruktionsverfahren bilden nach unterschiedlichen Vorgehensweisen erste Lösungsmöglichkeiten für das kapazitierte Laubharkproblem, zumal die Kapazitätsrestriktionen dabei berücksichtigt werden. Im Folgenden sollen nun zulässige Nachfolgerfunktionen s dahingehend verändert werden, dass der Zielfunktionswert $K(s)$ sich schrittweise verbessert. Die Veränderungen sollen durch möglichst einfache Tauschschritte erfolgen. Um dabei die Zulässigkeit der Lösungen zu erhalten, müssen gewisse Einschränkungen an den Tauschmöglichkeiten beachtet werden. Insbesondere muss garantiert werden, dass es sich nach einer Änderung an einer (zulässigen) Nachfolgerfunktion wieder um eine solche handelt.

3.3.1 Lösungsverbesserung durch einfachen Nachfolgertausch innerhalb eines Clusters

Bei diesem Verbesserungsverfahren wird in jeder Iteration für *genau einen* Knoten $a \in C$ eines bestimmten Clusters C die Zuweisung $s(a)$ gemäß einer zulässigen Nachfolgerfunktion s geändert, indem dieser Nachfolgerknoten von a durch einen anderen Knoten $b \in C$ ersetzt wird. Dabei werden sämtliche Tauschmöglichkeiten systematisch durchlaufen, wobei stets geprüft wird, ob die neue Abbildung $s^* : V \rightarrow V$ wiederum eine (zulässige) Nachfolgerfunktion ist und den Zielfunktionswert verbessert.⁴⁹

Diese lokale Suche nach einem „Ersatzknoten“ b für $s(a)$ (d.h. Knoten a ist vorgegeben) kann auf zweierlei Weise implementiert werden: Zum einen können zunächst sämtliche „Ersatzkandidaten“ bestimmt und daraus der Beste im Sinne einer Zielwertverbesserung ausgewählt werden (*Best-Fit-Variante*), zum anderen könnte der erste gefundene Knoten b , der eine Zielfunktionswertverbesserung liefert, als Ersatzknoten für $s(a)$ bestimmt werden (*First-Fit-Variante*).

Bei der „Globalisierung“ dieser lokalen Suche auf das Cluster C , indem *jeder* Knoten $a \in C$ einer lokalen Suche unterzogen wird, kann wiederum eine Best-Fit- oder First-Fit-Variante implementiert werden, um denjenigen Knoten a festzulegen, dessen Wert $s(a)$ durch einen zugehörigen Knoten $b \in C$ ersetzt wird. Schließlich ergibt sich pro Iteration *ein* neuer Nachfolger $b \in C$ zu *einem* Knoten $a \in C$ (falls existent). Diese Prozedur kann solange wiederholt werden, bis sich anhand dieser Kandidatensuche keine

⁴⁹ Hierzu siehe Algorithmus 15 im Anhang B.

Verbesserung mehr erreichen lässt.

Letztendlich wird diese Prozedur auf alle Cluster der durch s induzierten Clusterung von G angewendet.

3.3.2 Lösungsverbesserung durch mehrfachen Nachfolgertausch innerhalb eines Clusters

Dieses Verbesserungsverfahren tauscht innerhalb eines jeden Clusters C für jedes Feld a seinen Nachfolger $s(a)$ durch ein anderes Feld $b \in C$ in Analogie zur First-Fit-Variante des einfachen Nachfolgertausches. Dabei wird bei jeder Verbesserung bereits die Nachfolgerfunktion s entsprechend geändert und schon bei der weiteren Kandidatensuche innerhalb der jeweiligen Iteration als aktuelle Nachfolgerfunktion zu Grunde gelegt. Dadurch können sich nach Abarbeitung aller Knoten $a \in C$ mehrere Veränderungen an der ursprünglichen Nachfolgerfunktion ereignet haben.⁵⁰

3.3.3 Lösungsverbesserung durch Einfügestrategie

Dieses Verbesserungsverfahren ist nicht auf die einzelnen Cluster eingegrenzt, sondern sucht für jeden Knoten $a \in V$ geeignete Nachfolger in der nächsten Nachbarschaft und tauscht diesen ein. D.h. für jeden Knoten $a \in V$ werden zunächst die Nachbarn b in G bestimmt ($[a, b] \in E$). Dann werden nacheinander diese Nachbarn als Nachfolger von a eingesetzt. Dabei findet wieder die Überprüfung auf Gültigkeit und ggf. eine Güteprüfung statt. Ist der Zielfunktionswert geringer, wird die geänderte Nachfolgerfunktion übernommen und in der nächsten Iteration verwendet (First-Fit-Variante).⁵¹ Verbildlicht passiert der folgende Schritt:

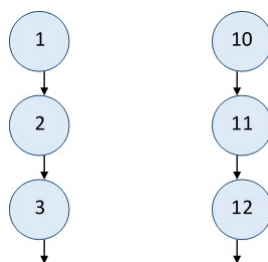


Abbildung 15: Ausgangslage

In dem oberen Beispiel sind zwei Harkfolgen zu sehen, die zu verschiedenen Clustern gehören. Das Verfahren ersetzt nun den Nachfolger eines Knotens durch einen anderen, wenn die Nachfolgerbedingungen erhalten bleiben und der Zielfunktionswert dadurch geringer wird. Die Vorgänger dieses Knotens bleiben seine Vorgänger, werden also in

⁵⁰ Eine algorithmische Beschreibung findet sich als Algorithmus 16 im Anhang B.

⁵¹ Alternativ lässt sich auch eine Best-Fit-Variante implementieren.

die neue Harkfolge mitgenommen. In diesem Beispiel bekommt der Knoten 2 als neuen Nachfolger den Knoten 11 bzw. wird der Knoten 2 nun Vorgänger des Knotens 11, also $s(2) = 11$ statt $s(2) = 3$.

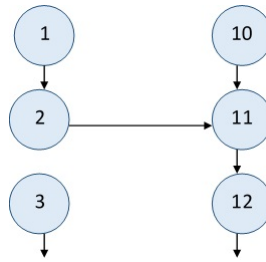


Abbildung 16: Neue Clusterbildung

Die Harkfolge ändert sich somit dahingehend, dass der Knoten 11 nun zwei Vorgänger hat, der Knoten 3 hat keine Vorgänger mehr. Die Nachbarschaftsbedingungen und die Eindeutigkeit der Zuweisungen bleiben weiterhin bestehen.⁵²

Diese Art des Harkfolgentaushes basiert auf dem sog. *Relocate-Verfahren* von Savelsbergh [7]. 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 [9] 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. Indem ein Knoten („Einfügeknoten“) einen neuen Nachfolger in einem anderen Cluster erhält, zieht dieser Knoten seine eigenen Vorgänger mit in das neue Cluster. Insbesondere werden dabei weitere Harkquellen dem erweiterten Cluster zugefügt. Anders als beim Tourenplanungsproblem wird dieser Einfügeknoten a aber nicht zwischen zwei Knoten platziert, sondern dem neuen Nachfolger „vorgelagert“.

3.3.4 Lösungsverbesserung durch Clustervereinigung

Es hat sich gezeigt, dass die Konstruktionsverfahren oft auch „kleine“ Cluster erzeugen (im Extremfall sogar einelementige Cluster), bei denen die kumulierte Harklaubmenge $M_s^*(h)$ im zugehörigen Hub h sehr klein ausfällt. Daher kann es – insbesondere auch im Hinblick auf den Transportprozess – opportun sein, benachbarte Cluster zusammenzulegen, wobei allerdings die Kapazitätsrestriktionen im Auge zu halten sind.

Ein Prozess zur Clustervereinigung könnte folgendermaßen aussehen⁵³:

In einem ausgewählten Cluster C_i mit dem Hub h_i und der kumulierten Laubharkmenge $M_s^*(h_i)$ wird jeder Knoten $a \in C_i$ daraufhin geprüft, ob dieser in G zu einem Knoten

⁵² Eine algorithmische Beschreibung findet sich als Algorithmus ?? im Anhang B.

⁵³ Hierzu siehe Algorithmus 18 im Anhang B.

3 Lösungsstrategien für das Laubharkproblem

$b \in V \setminus C_i$ benachbart ist. Das Cluster eines gefundenen Nachbarknotens b sei C_j mit dem Hub h_j und der kumulierten Laubharkmenge $M_s^*(h_j)$. Falls die Vereinigung dieser beiden Cluster zulässig ist, d.h. die gemeinsame Laubharkmenge erfüllt die Kapazitätsbedingung, wird ein temporärer Harkplan entworfen, bei dem alle Knoten in C_i auf Knoten a als neues Hub ausgerichtet werden und die Laubharkmenge $M_s^*(h_i)$ von dort zum Knoten b und weiter zum Hub h_j geharkt wird. Dieser neue Harkplan wird daraufhin überprüft, ob er kostengünstiger als der bisherige Harkplan ist. Falls dem so ist, wird die entsprechende Nachfolgerfunktion s_{neu} , die sich insbesondere dadurch kennzeichnet, dass sie ein Cluster weniger induziert, übernommen; andernfalls wird die Suche nach „Übergangsknoten“ a und b fortgesetzt, bis sich kein Knotenpaar mehr finden lässt.

Zur Auswahl von geeigneten Clustern zur Vereinigung könnte eine Schranke für die kumulierte Laubharkmenge gesetzt werden, sodass nur hinreichend kleine Cluster einem potentiellen Vereinigungsprozess zugeführt werden.

Literaturverzeichnis

- [1] Clarke G, Wright JW: *Scheduling vehicles from central delivery depot to a number of delivery points*. Operations Research Quarterly 12:568-581, 1964.
- [2] Domschke W, Drexl A: *Logistik: Standorte*; 4. Auflage. Oldenbourg 2014.
- [3] Domschke W, Scholl A: *Logistik: Rundreisen und Touren*; 5. Auflage. Oldenbourg 2010.
- [4] Felker E: *Analyse der unproduktiven Wegezeiten beim Laubharkproblem*. Bachelorarbeit im Studiengang Angewandte Mathematik der Fachhochschule Bielefeld, 2016.
- [5] Kruse H-J: *Mathematische Modellierung des Laubharkproblems als spezielle Ausprägung von allgemeinen Entsorgungsprozessen*. In: Kruse H-J, Lask T (Hrsg.): *Angewandte mathematische Modellierung und Optimierung - Ausgewählte Modelle, Methoden, Fallstudien*. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik, Band 4, Fachhochschule Bielefeld, 2016 [ISSN 2196-6192] S. 62-88.
- [6] Kruse H-J, Spenst N, Derdau R: *Heuristische Lösungsstrategien für das Laubharkproblem*. In: Kruse H-J, Lask T (Hrsg.): *Angewandte mathematische Modellierung und Optimierung - Ausgewählte Modelle, Methoden, Fallstudien*. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik, Band 4, Fachhochschule Bielefeld, 2016 [ISSN 2196-6192] S. 89-116.
- [7] Savelsbergh MWP: *The vehicle routing problem with time windows: Minimizing route duration*. Journal on Computing, 4:146-154, 1992.
- [8] Spenst N: *Entwicklung und Implementierung von effizienten Lösungsverfahren für ein spezielles Entsorgungsproblem*. Masterarbeit im Studiengang Optimierung und Simulation der Fachhochschule Bielefeld, 2015.
- [9] Van Breedam A: *Comparising descent heuristics and metaheuristics for the vehicle routing problem*. Computers and Operations Research, 28:289-315, 2001.

Anhang

Anhang A: Glossar

Anhang B: Algorithmen

Anhang C: Lösungsstrategien an Beispielen

A Glossar

Bezeichner	Bedeutung	Seite
$a \in \mathbb{N}$	Feld eines Gartens	4
$G = (V, E)$	Graph , durch den der Garten modellhaft beschrieben wird	7
V	Knotenmenge des Graphen G ; entspricht den Feldern des durch G modellierten Gartens	7
E	Kantenmenge des Graphen G ; beschreibt die Nachbarschaft der Felder des durch G modellierten Gartens	7
C	Bewertungsmatrix eines kantenbewerteten Graphen	8
D	Entfernungsmatrix eines kantenbewerteten Graphen	8
$s : V \rightarrow V$	Nachfolgerfunktion , wenn die <i>Nachbarschaftsbedingung</i> (2.9) gilt und der durch $s \rightarrow$ induzierte <i>Teilgraph</i> \vec{G}_s zyklensfrei ist	14
$\vec{G}_s = \langle V, \vec{E}_s \rangle$	durch die \rightarrow <i>Nachfolgerfunktion</i> s induzierter Teilgraph von $G = (V, E)$ mit $\vec{E}_s = \{ \langle a, s(a) \rangle \mid a \in V \wedge a \neq s(a) \}$	14
$h \in V$	Haufenfeld oder Hub \equiv Senke des durch die \rightarrow <i>Nachfolgerfunktion</i> s \rightarrow induzierten <i>Teilgraphen</i> \vec{G}_s , d.h. $s(h) = h$	14
$Hub(s)$	Hubmenge \equiv Menge aller \rightarrow <i>Haufenfelder</i> (\rightarrow <i>Hubs</i>), welche durch die \rightarrow <i>Nachfolgerfunktion</i> s induziert werden	15
$q \in V$	Harkquelle \equiv Quelle des durch die \rightarrow <i>Nachfolgerfunktion</i> s \rightarrow induzierten <i>Teilgraphen</i> \vec{G}_s	15
$Q(s)$	Harkquellenmenge \equiv Menge aller \rightarrow <i>Harkquellen</i> , welche durch die \rightarrow <i>Nachfolgerfunktion</i> s induziert werden	15
$K \in V$	Kompostfeld \equiv Kompoststelle des Gartens	20
$M(a)$	initiale Laubmenge in \rightarrow <i>Feld</i> a ; entspricht der Knotenbewertung des \rightarrow <i>Graphen</i> G	5
$M^*(a)$	aktuelle Laubmenge in \rightarrow <i>Feld</i> a ; gibt die (zwischenzeitlich) kumulierte Laubmenge in Feld a zum Zeitpunkt des Harkens von Feld a zu einem Nachbarfeld an	9
M_k	kritische Laubharkmenge ; entspricht der Laubmenge, die noch mit einem einzigen Harkzug bewegt werden kann	10
$\overline{M}, \overline{M}(a)$	maximale Laubmenge (von \rightarrow <i>Feld</i> a); entspricht der Laubmenge, die maximal im Feld a angehuft werden kann	11
$M_s^* : V \rightarrow \mathbb{R}_{\geq}$	die durch die \rightarrow <i>Nachfolgerfunktion</i> s induzierte Anhufungsfunktion gem Bildungsvorschrift (2.10)	15,23

Bezeichner	Bedeutung	Seite
$\alpha_H, \alpha_H(a, b)$	Harkaufwandsfaktor oder Harkparameter	9
$HA(a, b)$	einzelner Harkaufwand ; entspricht dem Aufwand für das Harken der aktuellen Laubmenge von Feld a in ein benachbartes Feld b	8f
$HA_\Sigma(s)$	Harkaufwandsumme ; entspricht dem Gesamtaufwand des produktiven Harkprozesses, der durch die \rightarrow <i>Nachfolgerfunktion</i> s beschrieben wird	16
$\alpha_W, \alpha_W(a, q)$	Wegeaufwandsfaktor oder Wegeparameter	19
$W(s)$	unproduktive Weglänge ; entspricht der Gesamtlänge aller unproduktiven Wege, welche durch die \rightarrow <i>Nachfolgerfunktion</i> s hervorgerufen werden	19
$WA_\Sigma(s)$	unproduktiver Wegeaufwand ; entspricht dem Gesamtaufwand für unproduktive Wege des durch die \rightarrow <i>Nachfolgerfunktion</i> s beschriebenen Harkprozesses	19f
α_T	Transportaufwandsfaktor oder Transportparameter	21
\bar{T}	maximales Ladevolumen des Transportmittels	21
γ	variabler Ladeaufwand \equiv variabler Aufwandsanteil für Auf- und Abladen einer Laubmenge	21
σ	fixer Ladeaufwand \equiv fixer Aufwandsanteil für Auf- und Abladen einer Laubmenge	21
$TA(h)$	Transportaufwand von \rightarrow <i>Hub</i> h zum \rightarrow <i>Kompostfeld</i> K ; entspricht dem Aufwand für das Transportieren der kumulierten Laubmenge von \rightarrow <i>Hub</i> h zum \rightarrow <i>Kompostfeld</i> K , einschließlich der Aufwände für das Auf- und Abladen	21
$TA_\Sigma(s)$	Transportaufwandsumme ; ergibt sich als gesamter Transportaufwand aller Touren zwischen allen \rightarrow <i>Hubs</i> bzgl. \rightarrow <i>Nachfolgerfunktion</i> s und \rightarrow <i>Kompostfeld</i> , einschließlich aller Aufwände für das Auf- und Abladen	22
$M_s^{**}(h)$	eine nach entsprechend vielen Pendeltouren verbleibende Laubrestmenge beim \rightarrow <i>Hub</i> h	22
$\pi : V \leftrightarrow V$	Permutation zur Ausrichtung einer \rightarrow <i>Nachfolgerfunktion</i> s	19
$R(a)$	Erreichbarkeitsmenge ; enthält alle Knoten $i \in V$, von denen aus der Knoten a entlang einer Pfeilfolge im \rightarrow <i>induzierten Teilgraph</i> \vec{G}_s erreichbar ist	15,23
$C(h)$	Cluster ; entspricht der \rightarrow <i>Erreichbarkeitsmenge</i> einer Senke h des \rightarrow <i>induzierten Teilgraphen</i> \vec{G}_s	23
$\Lambda(G)$	Lösungsraum des Laubharkproblems	23
K	Kostenfunktion als Zielfunktion des Laubharkproblems	23

B Algorithmen

Lfd. Nr.	Bezeichnung	Seite
Algorithmus 1:	Prüfung der Bedingungen einer Nachfolgerfunktion	43
Algorithmus 2:	Zulässigkeitsprüfung für Nachfolgerfunktion	44
Algorithmus 3:	Bestimmung einer Erreichbarkeitsmenge	44
Algorithmus 4:	Clusterbestimmung	45
Algorithmus 5:	Berechnung des Harkaufwandes	45
Algorithmus 6:	Berechnung des unproduktiven Wegeaufwandes	46
Algorithmus 7:	Ausrichtung einer Nachfolgerfunktion	46
Algorithmus 8:	Berechnung von Hark- und Wegeaufwand anhand einer ausgerichteten Nachfolgerfunktion	47
Algorithmus 9:	Berechnung von Transportaufwänden	48
Algorithmus 10:	Zickzack-Strategie	49
Algorithmus 11:	Clusterbestimmung mit NW-Orientierung	50
Algorithmus 12:	Huborientiertes Konstruktionsverfahren (TS)	51
Algorithmus 13:	Huborientiertes Konstruktionsverfahren (BS)	52
Algorithmus 14:	Hubausrichtung	53
Algorithmus 15:	Lokales einfaches Nachfolgertauschverfahren	54
Algorithmus 16:	Mehrfaches Nachfolgertauschverfahren	55
Algorithmus 17:	Verbesserung durch Einfügestrategie	56
Algorithmus 18:	Verbesserung durch Clustervereinigung	56

Algorithmus 1 (Prüfung der Bedingungen einer Nachfolgerfunktion):**Vorgabe:** Graph $G = [V, E]$, Abbildung $s : V \rightarrow V$.**Schritt 1:** (Initialisierung)Setze: $H := \emptyset; N := \emptyset; Q := \emptyset$.Für alle $i \in V$ setze: $\delta(i) := 0$.**Schritt 2:** (Bestimmung der Harkquellen)Für alle $i \in V$ führe aus:Falls $i \neq s(i)$, setze: $\delta(s(i)) := \delta(s(i)) + 1$.Für alle $i \in V$ führe aus:Falls $\delta(i) = 0$, setze: $Q := Q \cup \{i\}$ Falls $Q = \emptyset$, terminiere! ⁵⁴**Schritt 3:** (Prüfe Nachbarschaftsbedingung)Für alle $i \in V$ führe aus:Falls $[i, s(i)] \notin E$, terminiere! ⁵⁵**Schritt 4:** (Auswahl einer Harkquelle)Wähle $k \in Q$.⁵⁶Setze: $Q := Q \setminus \{k\}$.**Schritt 5:** (Prüfe Zyklenfreiheit)Falls $s(k) \in N$, terminiere! ⁵⁷Falls $s(k) = k$, führe aus: $H := H \cup \{k\}$.Falls $Q = \emptyset$, terminiere! ⁵⁸

Gehe zu Schritt 4.

 $N := N \cup \{k\}$.Setze: $\delta(s(k)) := \delta(s(k)) - 1$.Falls $\delta(s(k)) \neq 0$, führe aus:Falls $Q = \emptyset$, terminiere! ⁵⁹

Gehe zu Schritt 4.

Setze: $k := s(k)$ und gehe zu Schritt 5.

Nach Überprüfung aller Knoten enthält die Menge H alle Hubs (Haufenfelder), die Menge N alle Nicht-Hubs für den durch die Nachfolgerfunktion s beschriebenen Harkprozess.

⁵⁴ Der durch die Abbildung s beschriebene Digraph besitzt keine Quelle und ist somit nicht zyklensfrei.

⁵⁵ Die Nachbarschaftsbedingung wird verletzt.

⁵⁶ Das Auswahlkriterium ist freigestellt; es könnte beispielsweise ein Schlangen- oder Stapelspeicher zugrundegelegt werden (d.h. FIFO- oder LIFO-Prinzip).

⁵⁷ Der durch die Abbildung s beschriebene Digraph enthält einen Zyklus.

⁵⁸ Die Überprüfung ist abgeschlossen, indem sämtliche Knoten des Digraphen untersucht worden sind.

⁵⁹ Widerspruch zur Zyklenfreiheit.

Algorithmus 2 (Zulässigkeitsprüfung für Nachfolgerfunktion):

Vorgabe: Graph $G = [V, E]$ mit Knotenbewertung $M : V \rightarrow \mathbb{R}_{\geq}$, Kapazitätsfunktion $\overline{M} : V \rightarrow \mathbb{R}_{\geq}$, Nachfolgerfunktion $s : V \rightarrow V$, Quellenmenge $Q \subset V$, Eingangsvalenzen $\delta(i)$ aller Knoten $i \in V$.⁶⁰

Schritt 1: (Vorprüfung)

Für alle $a \in V$ führe aus:

Falls $M(a) > \overline{M}(a)$, terminiere.⁶¹

Schritt 2: (Auswahl einer Harkquelle)

Falls $Q = \emptyset$, terminiere.

Wähle $k \in Q$.

Setze: $Q := Q \setminus \{k\}$, $M_{kum}(k) := M(k)$.

Schritt 3: (Zulässigkeitsprüfung)

Falls $s(k) = k$, gehe zu Schritt 2.

Falls $M_{kum}(k) + M_{kum}(s(k)) > \overline{M}(s(k))$, terminiere.⁶²

Setze: $M_{kum}(s(k)) := M_{kum}(s(k)) + M_{kum}(k)$, $\delta(s(k)) := \delta(s(k)) - 1$.

Falls $\delta(s(k)) \neq 0$, gehe zu Schritt 2.

Setze: $k = s(k)$.

Gehe zu Schritt 3.

Für eine zulässige Nachfolgerfunktion s ist mit M_{kum} die zugehörige Anhäufungsfunktion bestimmt. Das verwendete Suchprinzip entspricht dem sukzessiven Entblättern eines Baumes. Zur Bestimmung der kumulierten Laubmenge für nur einen ausgewählten Knoten a bietet sich auch das umgekehrte Suchprinzip an; hierzu vgl. Algorithmus 3.

Algorithmus 3 (Bestimmung einer Erreichbarkeitsmenge):

Vorgabe: Nachfolgerfunktion $s : V \rightarrow V$, bestimmter Knoten $a \in V$.

Schritt 1: (Initialisierung)

Setze $K := \{a\}$, $R := \{a\}$.

Schritt 2:

Wähle $k \in K$.

Setze: $K := K \setminus \{k\}$.

Für alle $i \in V \setminus R$ führe aus:

Falls $s(i) = k$, setze: $K := K \cup \{i\}$, $R := R \cup \{i\}$.

Falls $K = \emptyset$, terminiere;

andernfalls gehe zu Schritt 2.

⁶⁰ Zur Bestimmung von Q und der Eingangsvalenzen $\delta(i)$ vgl. Schritt 2 in Algorithmus 1.

⁶¹ Der Initialzustand ist bereits unzulässig.

⁶² Die Nachfolgerfunktion s ist nicht zulässig bzgl. \overline{M} .

Die Menge R enthält sämtliche Knoten, von denen aus im Digraph \vec{G}_s der ausgezeichnete Knoten a über eine Pfeilfolge in \vec{G}_s erreichbar ist; es handelt sich somit um die sog. Erreichbarkeitsmenge $R(a)$. Die Bestimmung der durch die Nachfolgerfunktion s induzierten kumulierten Laubmenge im Knoten a ergibt sich durch Aufsummierung $M^s(a) := \sum_{i \in R(a)} M(i)$. Diese Aufsummierung ließe sich ohne Weiteres auch im Algorithmus 3 integrieren.

Algorithmus 4 (Clusterbestimmung):

Vorgabe: Nachfolgerfunktion s , Hubmenge H , Quellenmenge Q .⁶³

Schritt 1: (Initialisierung)

Für alle $h \in H$ setze: $C[h] := \{h\}$.⁶⁴

Schritt 2:

Wähle $k \in Q$.

Setze: $Q := Q \setminus \{k\}, T := \{k\}$.

Schritt 3:

Falls $s(k) \in H$, führe aus:

Setze: $C[s(k)] := C[s(k)] \cup T$.

Falls $Q = \emptyset$, terminiere; andernfalls gehe zu Schritt 2.

Setze: $k := s(k), T := T \cup \{k\}$.

Gehe zu Schritt 3.

Die Menge $C[h]$ enthält diejenigen Knoten (Felder), von denen gemäß Nachfolgerfunktion s zum Hubfeld h geharkt wird, wobei das Hubfeld h selbst auch zum Cluster $C[h]$ gehört.⁶⁵ Diese explizite Clusterbestimmung ließe sich ohne Weiteres auch in den Algorithmus 1 integrieren.

Algorithmus 5 (Berechnung des Harkaufwandes):

Vorgabe: Graph $G = [V, E]$, Knotenbewertung $M : V \rightarrow \mathbb{R}_{\geq}$, Kantenbewertung $\alpha_H : E \rightarrow \mathbb{R}_{\geq}$, zulässige Nachfolgerfunktion s mit zugehöriger Anhäufungsfunktion M_s^* .⁶⁶

Schritt 1: (Initialisierung)

Setze: $HA_{\Sigma}(s) := 0$.

Schritt 2: (Berechnung)

Für alle $a \in V$ führe aus:

Falls $a \neq s(a)$, setze: $HA_{\Sigma}(s) := HA_{\Sigma}(s) + HA(a, s(a))$.⁶⁷

⁶³ Die Mengen H und Q lassen sich gemäß Algorithmus 1 bestimmen.

⁶⁴ Alternativ könnte auch die Setzung $C[h] := \emptyset$ implementiert werden, wenn die Cluster $C[h]$ die Hubs h selbst nicht beinhalten sollen.

⁶⁵ Um diese reflexive Zuweisung auszuschließen, müsste die Setzung entsprechend geändert werden.

⁶⁶ Zur Prüfung der Zulässigkeit von s sowie Ermittlung der Anhäufungsfunktion siehe Algorithmus 2.

⁶⁷ Die Einzelaufwände $HA(a, s(a))$ ergeben sich gemäß Harkaufwandsformel (2.6), (2.7) oder (2.8).

Der Wert $HA_{\Sigma}(s)$ gibt den reinen Harkaufwand des durch die vorgegebene Nachfolgerfunktion s definierten Harkprozessen an (ohne Berücksichtigung unproduktiver Wege).

Algorithmus 6 (Berechnung des unproduktiven Wegeaufwandes):

Vorgabe: Nachfolgerfunktion s , Entfernungsmatrix $D = (d_{ij})_{i,j \in V}$ des bewerteten Graphen $G = [V, E, c]$, Eingangsvalenzen $\delta(i)$ aller Knoten $i \in V$, Quellenmenge Q sowie ein aktuelles Feld $a \in V$.⁶⁸

Schritt 1: (Initialisierung)

Setze $u := 0$.

Schritt 2:

Falls $Q = \emptyset$, terminiere!

Bestimme $k \in Q$ mit $d_{ak} = \min\{d_{ai} \mid i \in Q\}$.⁶⁹

Setze: $Q := Q \setminus \{k\}$, $u := u + d_{ak}$.

Schritt 3:

Falls $s(k) = k$, führe aus:

Setze: $a := k$.

Gehe zu Schritt 2.

Setze $\delta(s(k)) := \delta(s(k)) - 1$.

Falls $\delta(s(k)) = 0$, führe aus:

Setze: $k := s(k)$.

Gehe zu Schritt 3.

Setze: $a := k$.

Gehe zu Schritt 2.

Der Ergebniswert u gibt die Gesamtlänge aller unproduktiven Wege an, wobei eine Bewertung mittels speziellem Aufwandsfaktor noch nicht berücksichtigt ist.

Algorithmus 7 (Ausrichtung einer Nachfolgerfunktion):

Vorgabe: Kantenbewerteter Graph $G = [V, E, c]$ mit Entfernungsmatrix $D = (d_{ij})_{i,j \in V}$, Nachfolgerfunktion s , Quellenmenge Q .⁷⁰

Schritt 1: (Initialisierung)

Setze: $i := 0$.

Wähle $a \in V$ als Startknoten.

⁶⁸ Die Eingangsvalenzen $\delta(i)$ inkl. Menge Q ergeben sich gemäß Schritt 2 von Algorithmus 1.

⁶⁹ Dieses spezielle Suchkriterium vom Greedy-Typ kann im Sinne der Optimierung des Gesamtaufwandes durch andere Suchkriterien ersetzt werden. Allgemein könnte man hier zunächst auch schreiben: Wähle $k \in Q$.

⁷⁰ Die Berechnung der Entfernungsmatrix wird mittels einem Kürzeste-Wege-Verfahren vollzogen, z.B. dem Dijkstra-Algorithmus. Zur Bestimmung der Menge $Q = Q(s)$ der Harkquellen siehe Schritt 2 in Algorithmus 1.

Schritt 2:

Falls $Q = \emptyset$, terminiere.

Bestimme $k \in Q$ mit $d_{ak} = \min\{d_{aj} \mid j \in Q\}$.

Setze: $i := i + 1, \pi(i) := k, Q := Q \setminus \{k\}$.

Schritt 3:

Falls $k = s(k)$, führe aus:

Setze: $a := k$.

Gehe zu Schritt 2.

Falls $\delta(s(k)) \neq 0$, führe aus:

Setze: $a := s(k)$.

Gehe zu Schritt 2.

Setze: $k := s(k), i := i + 1, \pi(i) := k$.

Gehe zu Schritt 3.

Die Permutation $\pi : V \rightarrow V$ liefert die ausgerichtete Nachfolgerfunktion s_π , indem die Werte $\pi(i)$ und $s(\pi(i))$ als die i -te Spalte in der Tabelle der ausgerichteten Nachfolgerfunktion s_π zu interpretieren sind. Es ist offensichtlich, dass sich das Auswahlkriterium für die jeweils nächste Harkquelle auch anders gestalten ließe. Zudem sollte ein Sekundärkriterium implementiert werden für den Fall, dass das Primärkriterium („Nächster Nachbar“) gegebenenfalls nicht eindeutig ist; hier könnte das „Prinzip des kleinsten Knotenindex“ angewendet werden.

Algorithmus 8 (Berechnung von Hark- und Wegeaufwand anhand einer ausgerichteten Nachfolgerfunktion):

Vorgabe: Ausgerichtete Nachfolgerfunktion s , Entfernungsmatrix $D = (d_{ij})_{i,j \in V}$ des bewerteten Graphen $G = [V, E, c]$.

Schritt 1: (Initialisierung)

Setze $\Sigma := 0$.

Schritt 2: (Berechnung)

Für $i = 1, \dots, n - 1$ führe aus:

Falls $i \neq s(i)$, setze: $\Sigma := \Sigma + HA(i, s(i))$.⁷¹

Falls $s(i) \neq i + 1$, setze: $\Sigma := \Sigma + \alpha_W(s(i), i + 1) \cdot d_{s(i), i+1}$.⁷²

Der Wert Σ gibt die kumulierten Hark- und Wegeaufwände an, die anhand der ausgerichteten Nachfolgerfunktion s als Harkprozess hervorgerufen werden. Es ist offensichtlich, dass sich ohne Weiteres auch die produktiven Harkaufwände und die unproduktiven

⁷¹ Die Einzelaufwände $HA(i, s(i))$ ergeben sich gemäß Harkaufwandsformel (2.6), (2.7) oder (2.8).

⁷² Bei konstantem Wegeaufwandsfaktor kann auch vereinfacht α_W geschrieben werden.

Wegeaufwände getrennt berechnen ließen.

Algorithmus 9 (Berechnung von Transportaufwänden):

Vorgabe: Entfernungsmatrix $D = (d_{ij})_{i,j \in V}$ des bewerteten Graphen $G = [V, E, c]$, Nachfolgerfunktion s , Hubmenge $Hub(s)$, Anhäufungsfunktion M_s^* , maximale Transportkapazität \bar{T} , Kompostfeld $K \in V$.

Schritt 1: (Initialisierung)

Setze: $TA_\Sigma := 0$.

Schritt 2: (Berechnung der Aufwände für Pendeltouren)

Für alle $h \in Hub(s)$ führe aus:

Falls $M_s^*(h) < \bar{T}$, setze: $M_s^{**}(h) := M_s^*(h)$;

andernfalls setze: $TA_\Sigma := TA_\Sigma + TA(h)$, $M_s^{**}(h) := M_s^*(h) - \left\lceil \frac{M_s^*(h)}{\bar{T}} \right\rceil \cdot \bar{T}$.

Falls $M_s^{**}(h) := 0$, setze: $Hub(s) := Hub(s) \setminus \{h\}$.

Schritt 3:

Falls $Hub(s) = \emptyset$, terminiere!

Setze: $T := 0$; $a := K$.

Schritt 4: (Zurechnung der Aufwände für Sammeltouren)

Bestimme $h \in Hub(s)$ mit $d_{ah} = \min\{d_{aj} \mid j \in Hub(s)\}$.

Falls $T + M_s^{**}(h) \leq \bar{T}$, führe aus:

Setze: $T := T + M_s^{**}(h)$, $TA_\Sigma := TA_\Sigma + \alpha_T \cdot d_{ah}$, $Hub(s) := Hub(s) \setminus \{h\}$.

Falls $Hub(s) \neq \emptyset$, führe aus:

Setze: $a := h$.

Gehe zu Schritt 4.

Setze: $TA_\Sigma := TA_\Sigma + \alpha_T \cdot d_{aK}$.

Gehe zu Schritt 3.

Im obigen Algorithmus sind die einfachen Transportaufwandsformeln (2.17) und (2.23) implementiert. Das Ergebnis TA_Σ gibt den danach ermittelten gesamten Transportaufwand bzgl. Nachfolgerfunktion s an.

Algorithmus 10 (Zickzack-Strategie):

Vorgabe: Knoten- und kantenbewerteter Graph $G = [V, E, c, M]$, Kapazitätsfunktion $\overline{M} : V \rightarrow \mathbb{R}_{>}$.

Schritt 1: (Initialisierung)

Für alle $a \in V$ setze: $M_{kum}(a) := M(a)$.

Schritt 2: (Aufstellen einer Zickzackfolge φ)

Für $i = 1, \dots, m$ führe aus:

Für $j = 1, \dots, n$ führe aus:

Setze: $a := (i - 1) \cdot n + j$.

Falls i gerade⁷³, setze: $\varphi(a) := a$;

andernfalls setze: $\varphi(a) := i \cdot n - j + 1$.

Schritt 3: (Ermittle Nachfolgerfunktion s mit Zickzackfolge)

Falls $\varphi(m \cdot n) \in V$, setze: $s(\varphi(m \cdot n)) := \varphi(m \cdot n)$

Für alle $a = 1, \dots, m \cdot n$ führe aus:

Falls $\varphi(a) \in V$, führe aus:

Falls $[\varphi(a), \varphi(a + 1)] \notin E$, setze: $s(\varphi(a)) := \varphi(a)$;

andernfalls führe aus:

Setze: $s(\varphi(a)) := \varphi(a + 1)$,

setze: $M_{kum}(\varphi(a + 1)) := M_{kum}(\varphi(a + 1)) + M_{kum}(\varphi(a))$.

Bei der Vorgabe der Knotenmenge V wird davon ausgegangen, dass die Knotenindizes a den Feldnummern (i, j) eines rechteckigen Gartens mit der Dimension $m \times n$ entsprechen: $a = a(i, j) = (i - 1) \cdot n + j$, wobei lediglich (nicht harkbare) Ausschlussfelder weggelassen sind (also Felder a mit $M(a) = -1$). Der Ergebnisvektor s stellt eine zulässige Nachfolgerfunktion von G dar, die entsprechend der Konstruktionsanweisung ein waagerechtes Zickzack-Harken nachahmt. Zudem ist mit φ bereits eine Ausrichtung von s gegeben, d.h. $\pi(a) = \varphi(a)$ (vgl. S. 19). Möglicherweise ist es opportun, auch Ausschlussfelder in den Ergebnisvektor s aufzunehmen, etwa durch die Setzung: $s(a) = 0 \Leftrightarrow a \notin V$.

Die obige einfache Zickzack-Strategie lässt sich ohne Weiteres auch auf spaltenmäßig unterteilte Teilmatrizen anwenden. Hierzu sei die Indexfolge $1, \dots, n$ wie folgt in Teilfolgen zerlegt: $1, \dots, n_1, n_1 + 1, \dots, n_2, \dots, n_{k-1} + 1, \dots, n_k, \dots, n_{K-1} + 1, \dots, n$. Entsprechend müsste der obige Schritt 2 wie folgt verallgemeinert werden:

Schritt 2':

Für $k = 1, \dots, K$ führe aus:

⁷³ Die Eigenschaft einer ganzen Zahl i , gerade zu sein, lässt sich äquivalent als $i \bmod 2 = 0$ angeben bzw. prüfen.

Für $i = 1, \dots, m$ führe aus:

Für $j = n_{k-1} + 1, \dots, n_k$ führe aus:

Algorithmus 11 (Clusterbestimmung mit NW-Orientierung):

Vorgabe: Knotenbewerteter Graph $G = [V, E, M]$, Kapazität \overline{M} .

Schritt 1: (Initialisierung)

Setze: $V^{akt} := V$.⁷⁴

Für alle $a \in V^{akt}$ setze: $s(a) := 0$.

Schritt 2:

Falls $V^{akt} := \emptyset$, terminiere!

Wähle $a \in V^{akt}$.⁷⁵

Setze: $s(a) := a$, $V^{akt} := V^{akt} \setminus \{a\}$, $M^* := M(a)$, $k := a$, $N := \emptyset$.⁷⁶

Schritt 3:

Für alle $b \in V^{akt}$ mit $[b, k] \in E \wedge s(b) = 0$ führe aus:

Falls $M^* + M(b) \leq \overline{M}$ führe aus:

Setze: $N := N \cup \{b\}$, $s(b) := k$, $M^* := M^* + M(b)$, $V^{akt} := V^{akt} \setminus \{b\}$.

Falls $N = \emptyset$, gehe zu Schritt 2.

Wähle $k \in N$.

Setze: $N := N \setminus \{k\}$.

Gehe zu Schritt 3.

Die Orientierung an der nordwestlichen Ecke bei der Bestimmung aufzunehmender Knoten entspricht dem Indexkriterium (3.5). Analog könnte auch eine andere Eckenorientierung implementiert werden. Die sich ergebende Nachfolgerfunktion s liefert eine initiale Clusterbildung, an der sich eine Hubzentrierung (vgl. Abschn. 3.1.1) und eine Hubausrichtung (vgl. Abschn. 3.1.2 bzw. Algorithmus 14) in den einzelnen Clustern anschließen sollte.

In der obigen Beschreibung könnte auf die redundante Einführung der Variablen k verzichtet werden, indem durchgängig die Variable a benutzt wird. Es soll hier lediglich zwischen dem originären Hub a des neu zu bildenden Clusters und den anderen Clusterknoten unterschieden werden.

⁷⁴ V^{akt} ist als geordnete Liste anzulegen, worin die Knoten nach aufsteigenden Indizes geordnet sind. Speziell wird jeweils das erste Element entnommen.

⁷⁵ Da V^{akt} eine geordnete Liste ist, wird der Knoten in V^{akt} mit dem kleinsten Index gewählt, also unter den nördlichsten Feldern das westlichste.

⁷⁶ Speziell wird N als Schlangenspeicher (FIFO) angelegt.

Algorithmus 12 (Huborientiertes Konstruktionsverfahren (TS)):

Vorgabe: Knotenbewerteter Graph $G = [V, E, M]$, Kapazität \overline{M} .

Schritt 1: (Initialisierung)

Setze: $V^{akt} := V$.

Schritt 2:

Falls $V^{akt} := \emptyset$, terminiere!

Bestimme $a \in V^{akt}$ mit $M(a) = \max\{M(b) \mid b \in V^{akt}\}$.

Setze: $s(a) := a, V^{akt} := V^{akt} \setminus \{a\}, M^* := M(a), k := a$.

Schritt 3:

Setze: $N := \emptyset$.

Für alle $b \in V^{akt}$ mit $[b, k] \in E \wedge M^* + M(b) \leq \overline{M}$ setze: $N := N \cup \{b\}$.

Schritt 4:

Falls $N = \emptyset$, führe aus:

Falls $s(k) = k$, gehe zu Schritt 2;

andernfalls führe aus:

Setze: $k := s(k)$.

Gehe zu Schritt 3.

Schritt 5:

Bestimme $b^* \in N$ mit $M(b^*) = \min\{M(b) \mid b \in N\}$.

Falls $M(b^*) > M(k)$, setze: $k := s(k)$;

andernfalls führe aus:

Falls $M^* + M(b^*) \leq \overline{M}$, führe aus:

Setze: $s(b^*) := k, M^* := M^* + M(b^*), V^{akt} := V^{akt} \setminus \{b^*\}, k := b^*$.

Gehe zu Schritt 3.

Wenn das Minimalmengen-Kriterium in Schritt 5 durch das Maximalmengen-Kriterium ersetzt werden soll, muss eine erweiterte Kapazitätsprüfung erfolgen, sodass Schritt 5 folgende Form annimmt:

Schritt 5* :

Bestimme $b^* \in N$ mit $M(b^*) = \max\{M(b) \mid b \in N\}$.

Falls $M(b^*) > M(k)$, setze: $k := s(k)$;

andernfalls führe aus:

Falls $M^* + M(b^*) \leq \overline{M}$, führe aus:

Setze: $s(b^*) := k, M^* := M^* + M(b^*), V^{akt} := V^{akt} \setminus \{b^*\}, k := b^*$;

andernfalls führe aus:

Setze: $N := N \setminus \{b^*\}$.

Gehe zu Schritt 4.

Gehe zu Schritt 3.

Die Bestimmungskriterien in den Schritten 2 und 5 (bzw. 5*) sollten zwecks Eindeutigkeit um das Indexkriterium als Sekundärkriterium ergänzt werden.

Aufgrund der Übergänge zum jeweils aktuellen Knoten findet in den obigen Implementierungen des huborientierten Konstruktionsverfahrens eine Tiefensuche (TS) statt. Eine weitere Verfahrensvariante ergibt sich durch eine Breitensuche (BS):

Algorithmus 13 (Huborientiertes Konstruktionsverfahren (BS)):

Vorgabe: Knotenbewerteter Graph $G = [V, E, M]$, Kapazität \overline{M} .

Schritt 1: (Initialisierung)

Setze: $V^{akt} := V, N := \emptyset$. [N ist Stapelspeicher!]

Schritt 2:

Falls $V^{akt} := \emptyset$, terminiere!

Bestimme $a \in V^{akt}$ mit $M(a) = \max\{M(b) \mid b \in V^{akt}\}$.

Setze: $s(a) := a, V^{akt} := V^{akt} \setminus \{a\}, M^* := M(a), k := a$.

Schritt 3:

Setze: $N^{temp} := \emptyset, \mu_{max} := 0$.

Für alle $b \in V^{akt} \setminus N$ mit $[b, k] \in E \wedge M(b) \leq M(k)$ führe aus:

Setze: $N^{temp} := N^{temp} \cup \{b\}$.

Falls $M(b) > \mu_{max}$, setze: $\mu_{max} := M(b)$.

Setze: $\mu := \mu_{max}$.

Solange $\mu \geq 0 \wedge N^{temp} \neq \emptyset$, führe aus:

Bestimme $b \in N^{temp}$ mit $M(b) = \mu$.

Falls existent, setze: $N := N \cup \{b\}, s^{temp}(b) := k, N^{temp} := N^{temp} \setminus \{b\}$;

andernfalls setze: $\mu := \mu - 1$.

Schritt 4:

Falls $N = \emptyset$, gehe zu Schritt 2.

Bestimme $b \in N$. [Beachte: N ist Stapelspeicher!]

Setze $N := N \setminus \{b\}$.

Falls $M^* + M(b) \leq \overline{M}$, setze: $s(b) := s^{temp}(b), M^* := M^* + M(b), V^{akt} := V^{akt} \setminus \{b\}$.

Setze: $k := b$.

Gehe zu Schritt 3.

Man beachte, dass N als Stapelspeicher anzulegen ist. Die Bestimmungskriterien in den Schritten 2, 3 und 4 sollten zwecks Eindeutigkeit um das Indexkriterium ergänzt werden.

Algorithmus 14 (Hubausrichtung):**Vorgabe:** Graph $G = [V, E]$, Cluster $C \subseteq V$, Hub $h \in V$.**Schritt 1:** (Initialisierung)Setze: $L := \{h\}, s(h) := h, M := \emptyset$.**Schritt 2:** (Zuweisung)Wähle $a \in L$.Setze: $L := L \setminus \{a\}, M := M \cup \{a\}$.Für alle $b \in C \setminus (M \cup L)$ führe aus:Falls $[b, a] \in E$, setze: $L := L \cup \{b\}, s(b) := a$.Falls $L \neq \emptyset$, gehe zu Schritt 2;

andernfalls terminiere!

Der Ergebnisvektor s stellt eine auf das Cluster C eingeschränkte Nachfolgerfunktion dar, welche komplettiert wird, wenn der Algorithmus sukzessiv auf alle Cluster C_1, \dots, C_k eines Clusterings von G durchgeführt wird. Die Liste L muss ein Schlangenspeicher (FIFO) sein.

Während beim obigen Algorithmus die Knotenbearbeitung vom Hub zu den Harkquellen verläuft, beginnt der dazu alternative Algorithmus bei den Harkquellen und endet beim Hub:

Algorithmus 14* (Hubausrichtung):**Vorgabe:** Graph $G = [V, E]$, Cluster $C \subseteq V$, Hub $h \in V$.**Schritt 1:** (Initialisierung)Setze: $d_{max} := 0$.Für alle $a \in C$ führe aus:Bestimme die Entfernung d_{ah}^C vom Knoten a zum Hub h in C .Falls $d_{ah}^C > d_{max}$, setze: $d_{max} := d_{ah}^C$.**Schritt 2:** (Zuweisung)Für alle $a \in C$ mit $d_{ah}^C = d_{max}$ führe aus:Bestimme $b \in C$ mit $d_{bh}^C = d_{max} - 1 \wedge [a, b] \in E$.Setze: $s(a) := b$.Setze: $d_{max} := d_{max} - 1$.Falls $d_{max} > 0$, gehe zu Schritt 2;

andernfalls terminiere!

Algorithmus 15 (Lokales einfaches Nachfolgertauschverfahren):

Vorgabe: Knoten- und kantenbewerteter Graph $G = [V, E, M, c]$, Kapazitätsfunktion \overline{M} , Zielfunktion K , Nachfolgerfunktion s , ein durch s induziertes Cluster $C \subseteq V$, Knoten $a \in C$.

Schritt 1: (Initialisierung)

Berechne $K(s)$.

Setze: $K_{akt} := K(s)$, $k := s(a)$, $b_{best} := s(a)$.

Schritt 2: (Bestimmung einer neuen Zuweisung)

Für alle $b \in C \setminus \{a\}$ führe aus:

Falls $[a, b] \in E \wedge s(a) \neq b \wedge s(b) \neq a$, führe aus:

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

Prüfe, ob s eine Nachfolgerfunktion ist (Algorithmus 1).⁷⁷

Prüfe, ob s zulässig ist (Algorithmus 2).⁷⁸

Falls Prüfungsergebnis = *TRUE*, führe aus:

Berechne $K(s)$.

Falls $K(s) < K_{akt}$, setze: $K_{akt} := K(s)$, $b_{best} := b$.

Setze: $s(a) := k$.

Setze: $s(a) := b_{best}$.

Zu einem gegebenen Knoten $a \in C$ mit ursprünglichem Nachfolger $s(a)$ wird *ein* neuer Nachfolger b ermittelt, wodurch der Zielfunktionswert maximal verringert wird (falls existent); es handelt sich somit um die *Best-Fit-Variante*. Die *First-Fit-Variante* wäre dadurch gegeben, dass der Algorithmus bereits nach der ersten Verbesserung des Zielfunktionswertes abbricht.

Dieses Verfahren kann auf das Cluster C „globalisiert“ werden, indem das lokale Verfahren reihum auf jeden einzelnen Knoten $a \in C$ angewendet wird. Hierzu lässt sich wieder eine First-Fit- oder Best-Fit-Variante erstellen. Man mache sich allerdings klar, dass der Rechenaufwand bei einer „doppelten“ Best-Fit-Variante bedeutend höher als bei einer „doppelten“ First-Fit-Variante ausfällt⁷⁹, um dadurch lediglich einen einzigen Wert in der Nachfolgerfunktion zu ändern.

Das globalisierte Verfahren kann dann solange auf die jeweils neue Nachfolgerfunktion s angewendet werden, bis sich mit diesem Verfahren keine Zielfunktionsverbesserung mehr erzielen lässt oder eine Iterationsschranke erreicht ist.

⁷⁷ Insbesondere ist hierbei auf Zyklensfreiheit zu prüfen. Diese Prüfung kann ausgelassen werden, wenn Knoten a ein Hub ist.

⁷⁸ Diese Prüfung kann weggelassen werden, wenn die Kapazitätsfunktion \overline{M} konstant ist.

⁷⁹ Darüber hinaus ließen sich auch „gemischte“ Varianten implementieren (First-Fit/Best-Fit oder Best-Fit/First-Fit).

Algorithmus 16 (Mehrfaches Nachfolgertauschverfahren):

Vorgabe: Knoten- und kantenbewerteter Graph $G = [V, E, M, c]$, Kapazitätsfunktion \overline{M} , Zielfunktion K , Nachfolgerfunktion s , ein durch s induziertes Cluster $C \subseteq V$ mit Hub h .

Schritt 1: (Initialisierung)

Berechne $K(s)$.

Setze: $K_{akt} := K(s)$.

Schritt 2: (Bestimmung von neuen Zuweisungen)

Für alle $a \in C \setminus \{h\}$ führe aus:

Für alle $b \in C \setminus \{a\}$ führe aus:

Falls $[a, b] \in E \wedge s(a) \neq b \wedge s(b) \neq a$, führe aus:

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

Falls s keine (zulässige) Nachfolgerfunktion ist⁸⁰, setze: $s(a) := k$;

andernfalls führe aus:

Berechne $K(s)$.

Falls $K(s) < K_{akt}$, setze: $K_{akt} := K(s)$;

andernfalls setze: $s(a) := k$.

Im Gegensatz zum einfachen Nachfolgertausch (Algorithmus 15) wird die Nachfolgerfunktion s bei jeder Verbesserung des Zielfunktionswertes geändert, sodass sich nach dem systematischen Durchlauf über alle Knoten $a \in C \setminus \{h\}$ mehrere Zuweisungen $s(a)$ geändert haben können.

Durch Einfügung einer Sonderbehandlung für den Fall $a = h$ bzw. $s(a) = a$ kann das obige Verfahren für einen Durchlauf über alle $a \in C$ ergänzt werden:

Im Fall $a = s(a)$ führe aus:

Für alle $b \in C \setminus \{a\}$ führe aus:

Falls $[a, b] \in E$, führe aus:

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

Falls s nicht zulässig⁸¹, setze: $s(a) := a, s(b) := k$;

andernfalls führe aus:

Berechne $K(s)$.

Falls $K(s) < K_{akt}$, setze: $K_{akt} := K(s)$;

andernfalls setze: $s(a) := a, s(b) := k$.

⁸⁰ Hierzu siehe Prüfmodule beim einfachen Nachfolgertausch (Algorithmus 15). Dabei gelten dieselben Anmerkungen.

⁸¹ Im Falle einer konstanten Kapazitätsfunktion \overline{M} kann die Zulässigkeitsprüfung entfallen.

Algorithmus 17 (Verbesserung durch Einfügestrategie):

Vorgabe: Knoten- und kantenbewerteter Graph $G = [V, E, M, c]$, Kapazitätsfunktion \overline{M} , Zielfunktion K , Nachfolgerfunktion s .

Schritt 1: (Initialisierung)

Berechne $K(s)$.

Setze: $K_{akt} := K(s)$.

Schritt 2: (Bestimmung von neuen Zuweisungen)

Für alle $a \in V$ mit $s(a) \neq a$ führe aus:

Für alle $b \in V$ mit $[a, b] \in E$ führe aus:

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

Falls s eine zulässige Nachfolgerfunktion ist, setze: $s(a) := k$;

andernfalls führe aus:

Berechne $K(s)$.

Falls $K(s) < K_{akt}$, setze: $K_{akt} := K(s)$;

andernfalls setze: $s(a) := k$.

Eine Nachfolgerfunktion s mit verbessertem Zielfunktionswert induziert in der Regel neue Cluster. Da allerdings die Hubs bei der Suche nach besseren Zuweisungen ausgeschlossen werden, ändert sich nicht die Clusteranzahl. Der Durchlauf ist solange mit jeweils neuer Lösung zu wiederholen, bis sich keine Verbesserung mehr auftut oder eine Iterationsschranke erreicht ist.

Algorithmus 18 (Verbesserung durch Clustervereinigung):

Vorgabe: Knoten- und kantenbewerteter Graph $G = [V, E, M, c]$, konstante Kapazitätsschranke \overline{M} , Zielfunktion K , Nachfolgerfunktion s mit induzierten Clustern C_1, \dots, C_ℓ von G , zugehörigen Hubs h_1, \dots, h_ℓ und kumulierten Laubharkmengen $M_s^*(h_j)$ für $j = 1, \dots, \ell$; zudem Schranke M_{krit} für M_s^* .

Prozedur:

Für $j = 1, \dots, \ell$ führe aus:

Falls $M_s^*(h_j) \leq M_{krit}$ führe aus:

Für alle $a \in C_j$ führe aus:

Für alle $b \in V \setminus C_j$ mit $[a, b] \in E$ führe aus:

Setze: $k := b$.

Solange $k \neq s(k)$, setze: $k := s(k)$.⁸²

Setze: $h := k$.

Falls $M_s^*(h_j) + M_s^*(h) \leq M_{krit}$ führe aus:

Setze $s_{temp} := s$.

⁸² In dieser WHILE-Schleife wird das zum Knoten b gehörende Hub h ermittelt.

Falls $s(a) \neq a$, richte für s_{temp} alle $k \in C_j$ auf a aus.⁸³
 Setze $s_{temp}(a) := b$.
 Falls $K(s_{temp}) < K(s)$, führe aus:
 Setze: $s := s_{temp}$;
 Terminiere!

Die Schranke M_{krit} legt fest, ab wann ein Cluster „kritisch“ ist, d.h. „klein genug“, um für einen „Einschmelzungsprozess“ zugelassen zu werden. Für hinreichend großes M_{krit} können auch alle Cluster zur Auswahl gestellt werden.

Die obige Prozedurbeschreibung gilt nur für den Fall einer konstanten Kapazitätsfunktion \overline{M} . Eine Verallgemeinerung auf beliebige Kapazitätsfunktionen ist sehr viel aufwändiger auszuformulieren (und rechenintensiver). Darüber hinaus ist die obige Prozedur eine First-Fit-Variante, die bei der ersten Verbesserung abbricht (oder ohne Verbesserungsergebnis endet). Eine rechenintensivere Best-Fit-Variante könnte sowohl die Suche nach einem „besten ersten“ Übergangsknoten a als auch nach einem „besten zweiten“ Übergangsknoten b beinhalten. Die Prozedur ist solange zu wiederholen, bis sich keine Verbesserung mehr erzielen lässt oder eine Iterationsschranke erreicht ist.

⁸³ Hierzu vgl. Algorithmus 7.

C Lösungsstrategien am Beispiel

Auf das folgende Beispiel für einen Garten in Matrixdarstellung (vgl. Abb. 17) bzw. in Graphendarstellung (vgl. Abb. 18) sollen im Folgenden einige der in Kap. 3 behandelten Harkstrategien angewendet werden.

3	2	5	1	1	4	2	3	4	8	3	2	1
2	2	4		1	3	4	2	1	0	1	1	3
1	1	2	2			1	3	2	1	2	3	2
3	2	1	3	2	2	2	1	1	2	2	2	2
4	5	3	4	3	3	1	1	0	1	1	2	2
K	3	3	1	2			0	0	2	2	4	3

Abbildung 17: Beispiel für einen Garten in Matrixdarstellung

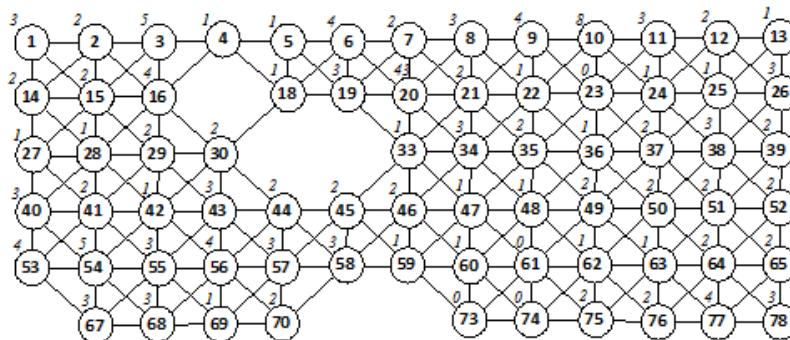


Abbildung 18: Beispiel für einen Garten in Graphendarstellung

Es werde der Einfachheit halber angenommen, dass sämtliche Kanten mit 1 bewertet sind und die dementsprechend ermittelten Entfernungen zwischen den Knoten in der zugehörigen Matrix $D = (d_{ij})$ hinterlegt sind ($i = 1, \dots, 6; j = 1, \dots, 13$). Zudem wird das südwestliche Feld als Kompostfeld festgelegt: $K = 66 \equiv (6, 1)$. Die Entfernung $d_{a,K}$ von den Knoten a zum Kompostfeld seien ebenfalls hinterlegt; dazu werde gesetzt: $d_{53,K} = d_{54,K} = d_{67,K} = 1$. Als Anfangsfeld für den Harkprozess wird der Knoten 1 bestimmt. Zudem wird von einer konstanten Kapazitätsfunktion \bar{M} ausgegangen und $\bar{M}(a) = 20$ für alle $a \in V$ festgelegt. Für die Transportkapazität gelte: $\bar{T} \geq 20$; somit wird zu jedem Hub nur eine Pendeltour benötigt.⁸⁴

⁸⁴ Einsparungsmöglichkeiten durch Sammeltouren werden in C.4 diskutiert.

Jede Ergebnisdarstellung besteht aus der jeweiligen Nachfolgerfunktion s in ausgerichteter Form und der Visualisierung des Harkprozesses, wobei die Pfeile die Harkrichtungen angeben, die entstandenen Cluster farbig hervorgehoben sind und in den Hubfeldern die kumulierten Laubharkmengen angegeben sind.

C.1 Zickzack-Strategien am Beispiel

C.1.1 Einfache Zickzack-Strategie

Die *einfache Zickzack-Strategie* liefert den folgenden Harkplan (Abb. 19) mit zulässiger Nachfolgerfunktion s :

a	1	2	3	4	5	6	7	8	9	10	11	12	13	26	25	24	23	22	21	20	19	18	16	15
$s(a)$	2	3	4	5	6	7	7	9	10	11	12	12	26	25	24	23	22	21	20	19	18	18	15	14
a	14	27	28	29	30	33	34	35	36	37	38	39	52	51	50	49	48	47	46	45	44	43	42	41
$s(a)$	27	28	29	30	30	34	35	36	37	38	39	52	51	50	50	48	47	46	45	44	43	42	41	40
a	40	53	54	55	56	57	58	59	60	61	62	63	64	65	78	77	76	75	74	73	70	69	68	67
$s(a)$	40	54	55	56	57	57	59	60	61	62	63	64	65	78	77	76	76	74	73	73	69	68	67	67

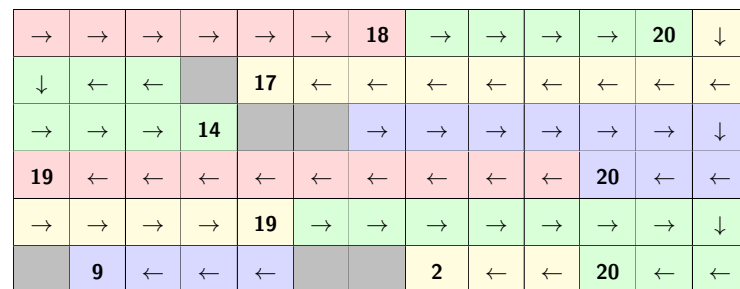


Abbildung 19: Harkplan durch einfache Zickzack-Strategie am Beispiel

Die gesamte Laubmenge des Gartens in Höhe von 158 [ME] werden dabei auf 10 Hubs zusammengeharkt. Dabei entsteht gemäß (2.6) ein produktiver Harkaufwand von $HA_{\Sigma}(s) = 517$ [ZE] bei Parametersetzung $\alpha_H = 1$ [ZE/ME].⁸⁵ Gemäß (2.14) ergibt sich als Summe der unproduktiven Wege zu $W(s) = 14$ [LE], die durch Gewichtung mit dem Wegeparameter α_W gemäß (2.15) einen unproduktiven Wegeaufwand $WA_{\Sigma}(s) = 14 \cdot \alpha_W$ [ZE] ergibt.⁸⁶ Für die Bestimmung des Transportaufwandes sind der Einfachheit halber ausschließlich Pendeltouren zwischen den Hubs und dem Kompostfeld zugelassen, deren Einzelaufwände gemäß (2.17) sich zum Gesamttransportaufwand

⁸⁵ Diese Standardsetzung für α_H erfolgt insofern o.B.d.A., da lediglich das Verhältnis zu den beiden anderen Parametern α_W und α_T das Gütemaß für eine Harkstrategie entscheidend beeinflusst.

⁸⁶ Hierbei ist zu beachten, dass die Übergänge von einem erreichten Hub zur nächsten Harkquelle hierbei in Zickzack-Richtung erfolgt. Bei Beachtung des „Nächster Nachfolger“-Prinzips ergäben sich 18 [LE].

$TA_{\Sigma} = 116 \cdot \alpha_T$ [ZE] aufsummieren (vgl. (2.21)). Demnach ergibt sich aus der einfachen Zickzack-Strategie ein Gesamtaufwand von $K(s) = 517 + 14 \cdot \alpha_W + 116 \cdot \alpha_T$ [ZE].

Durch Anwendung von *Hubzentrierung*⁸⁷ und *Hubausrichtung* ergibt sich der folgende Harkprozess (Abb. 20) mit zugehöriger Nachfolgerfunktion s , wobei sich der zugehörige Gesamtaufwand auf $K(s) = 254 + 47 \cdot \alpha_W + 114 \cdot \alpha_T$ [ZE] deutlich verringert, zumal $\alpha_W \leq 1$ angenommen werden kann.

a	1	2	14	27	15	28	29	30	16	18	19	7	6	5	4	3	40	41	42	57	56	55	53	54
$s(a)$	2	3	15	15	16	16	16	16	16	19	20	6	5	4	3	3	41	42	43	56	55	54	54	54

a	67	70	69	68	58	59	60	61	62	63	64	65	76	78	77	49	48	47	46	45	44	43	33	34
$s(a)$	68	69	68	68	59	60	61	62	63	77	77	77	77	77	77	48	47	46	45	44	43	43	34	35

a	35	36	37	26	11	8	9	10	12	13	25	24	23	22	21	20	50	39	51	52	38	73	74	75
$s(a)$	36	37	38	25	10	9	10	10	24	25	24	23	22	21	20	20	38	38	38	38	38	74	75	75

→	→	18	←	←	←	←	→	→	→	20	←	↘
→	→	14		→	→	17	←	←	←	←	←	←
↗	↗	↑	↖			→	→	→	→	→	20	←
→	→	→	19	←	←	←	←	←	←	↗	↑	↖
→	19	←	←	←	→	→	→	→	→	↘	↓	↘
	→	9	←	←			→	→	2	→	20	←

Abbildung 20: Harkplan durch einfache Zickzack-Strategie nach Hubausrichtung am Beispiel

C.1.2 Unterteilte Zickzack-Strategie

Die obige Gartenmatrix wird nun in drei Teilmatrizen unterteilt: $n_1 = 4, n_2 = 10$. Die Anwendung der *unterteilten Zickzack-Strategie* führt zu folgendem Harkprozess:

⁸⁷ Hierbei wird primär das *Maximalmengen-Kriterium* (3.2), sekundär das *Median-Kriterium* (3.3) und tertiär das *Indexkriterium* (3.5) angewendet; vgl. Abschn. 3.1.1.

a	1	2	3	4	16	15	14	27	28	29	30	43	42	41	40	53	54	55	56	69	68	67	5	6
$s(a)$	2	3	4	4	15	14	27	28	29	30	43	42	41	41	53	54	55	56	56	68	67	67	6	7

a	7	8	9	22	21	20	19	18	33	34	35	48	47	46	45	44	57	58	59	60	61	74	73	10
$s(a)$	8	9	22	21	21	19	18	18	34	35	48	47	46	45	44	57	58	58	60	61	74	73	73	11

a	11	12	13	26	25	24	23	36	37	38	39	52	51	50	49	62	63	64	65	78	77	76	75	70
$s(a)$	12	13	26	25	24	23	36	36	38	39	52	51	50	49	62	63	64	64	78	77	76	75	75	70

→	→	→	11	→	→	→	→	↓	→	→	→	↓
↓	←	←		8	←	←	17	←	↓	←	←	←
→	→	→	↓			→	→	↓	20	→	→	↓
↓	20	←	←	↓	←	←	←	←	↓	←	←	←
→	→	→	19	→	20	→	→	↓	→	→	19	↓
	7	←	←	2			2	←	13	←	←	←

Abbildung 21: Harkplan durch unterteilte Zickzack-Strategie am Beispiel

Die gesamte Laubmenge des Gartens in Höhe von 158 [ME] werden dabei auf 12 Hubs zusammengeharkt. Unter denselben Setzungen wie beim einfachen Zickzack-Harken entsteht nun ein Harkaufwand von $HA_{\Sigma}(s) = 570$ [ZE], ein Wegeaufwand $WA_{\Sigma}(s) = 19 \cdot \alpha_W$ [ZE] und ein Transportaufwand von $TA_{\Sigma} = 134 \cdot \alpha_T$ [ZE], insgesamt also ein Gesamtaufwand von $K(s) = 570 + 19 \cdot \alpha_W + 134 \cdot \alpha_T$ [ZE].

Da alle Teilaufwände bei obiger Lösung größer als beim einfachen Zickzack-Harken sind (vgl. Abb. 19), ist auch der Gesamtaufwand größer, und zwar unabhängig von den Harkparametern.

Durch Hubzentrierung und -ausrichtung dieser Startlösung (vgl. Abb. 22) erhält man eine deutlich verbesserte Lösung: $K(s) = 174 + 68 \cdot \alpha_W + 144 \cdot \alpha_T$. Unter der Annahme $\alpha_W \leq 1$ ist diese Lösung im Vergleich zur ebenfalls zentrierten und hubausgerichteten Lösung bei einfacher Zickzack-Strategie (vgl. Abb. 20) dann schlechter, wenn in etwa der Parameter $\alpha_T > 2,5$ ausfällt.

a	1	2	4	3	5	6	7	8	21	22	9	24	25	36	23	35	33	45	44	57	30	14	27	15
$s(a)$	2	3	3	3	6	7	8	9	9	9	9	10	11	23	10	34	45	58	58	58	16	15	15	16

a	18	19	20	47	48	34	46	58	43	29	40	41	42	28	16	53	67	56	55	54	69	68	73	61
$s(a)$	19	20	20	46	34	46	58	58	29	16	54	28	28	16	16	54	68	55	54	54	68	68	59	60

a	74	60	59	46	37	26	13	12	11	10	39	51	52	62	63	64	50	38	65	78	75	76	77	70
$s(a)$	60	59	59	37	38	12	12	11	10	10	38	38	38	50	50	50	38	38	77	77	76	77	77	70

→	→	11	←	→	→	→	→	17	20	←	←	←
→	→	20		→	→	8	↗	↑	↑	↖	↖	↖
↗	↗	↑	↖			↖	↖	←	↑	→	19	←
↘	↑	↖	↖	↘	↓	↖	←	↖	↗	↗	↑	↖
→	19	←	←	→	20	2	←	←	↗	↑	↖	↖
	→	7	←	2			↖	↖	→	→	13	←

Abbildung 22: Harkplan durch unterteilte Zickzack-Strategie nach Hubzentrierung und -ausrichtung am Beispiel

C.2 Clusterorientiertes Konstruktionsverfahren am Beispiel

Zur obigen Gartenmatrix und unter den angegebenen Bedingungen entsteht der folgende Harkplan (Abb. 23) durch Anwendung des *clusterorientierten Konstruktionsverfahrens* (vgl. Abschn. 3.2.2). Die Ausrichtung bei der Bestimmung neu aufzunehmender Knoten erfolgt in Ausrichtung auf die Nordwestecke, die auch Ausgangspunkt des Clusterbildungsprozesses darstellt. Somit kommt ausschließlich das Indexkriterium zum Einsatz. Nach der Bestimmung der Cluster werden diese einzeln einer Hubzentrierung und -ausrichtung unterzogen (vgl. Abschn. 3.1.1 bzw. Abschn. 3.1.2).

a	1	14	2	4	5	7	20	8	21	9	23	35	22	10	37	38	13	26	12	39	25	47	33	19
$s(a)$	2	2	3	5	6	6	6	9	9	10	10	22	10	10	24	24	12	12	11	25	11	34	19	6

a	16	27	28	15	3	40	42	30	18	6	43	29	41	53	54	67	55	44	57	45	59	60	70	58
$s(a)$	3	15	15	3	3	54	30	18	6	6	29	41	54	54	54	55	56	45	45	46	46	46	58	46

a	46	34	49	50	51	62	75	63	65	78	52	64	76	77	74	73	61	48	36	24	11	69	68	56
$s(a)$	34	34	63	63	63	63	63	77	78	78	64	77	77	77	61	61	48	36	24	11	11	56	56	56

→	→	20	→	→	20	←	→	→	20	20	←	←
↗	↗	↑		↗	↑	↖	↗	↗	↑	↑	↖	↖
↗	↑	↖	↗			↖	20	↑	↗	↑	↖	↖
↘	↓	↗	↖	→	→	↗	↑	↗	↘	↓	↖	↖
→	19	→	14	↗	↗	↑	↖	↑	→	↘	↓	↓
	↗	↗	↑	↗			↗	↑	↗	→	20	5

Abbildung 23: Harkplan durch clusterorientiertes Verfahren (NW-Orientierung) nach Hubausrichtung am Beispiel

Die gesamte Laubmenge des Gartens in Höhe von 158 [ME] werden dabei auf 9 Hubs zusammengeharkt. Unter denselben Setzungen wie bei den obigen Harkstrategien entsteht ein Harkaufwand von $HA_{\Sigma}(s) = 196 \cdot \alpha_H$ [ZE], ein Wegeaufwand $WA_{\Sigma}(s) = 65 \cdot \alpha_W$ [ZE] und ein Transportaufwand von $TA_{\Sigma} = 126 \cdot \alpha_T$ [ZE], insgesamt bei Setzung $\alpha_H = 1$ also ein Gesamtaufwand von $K(s) = 196 + 65 \cdot \alpha_W + 126 \cdot \alpha_T$ [ZE].

C.3 Huborientiertes Konstruktionsverfahren am Beispiel

Zur obigen Gartenmatrix und unter den angegebenen Bedingungen werden unterschiedliche Harkpläne nach dem *huborientierten Konstruktionsverfahren* entwickelt (vgl. Abschn. 3.2.3):

Zunächst wird bei der Knotenbestimmung für neue Hubs bzw. Harkvorgänger eine Tiefensuche (TS) vorgenommen. Dabei wird zum einen das Minimum-, zum zweiten das Maximummengen-Kriterium zur Auswahl der jeweiligen Harkvorgänger benutzt, wobei anschließend die Nachfolgerfunktionen auf die Hubs in den jeweiligen Clustern ausgerichtet werden (vgl. Abschn. 3.1.2). Beim letzten Harkplan wird eine Breitensuche (BS) vorgenommen. Als Sekundärkriterium kommt jeweils das Indexkriterium zum Einsatz.

C.3.1 Huborientiertes Konstruktionsverfahren via Tiefensuche

Bei der Auswahl der jeweiligen Harkvorgänger wird das Minimummengen-Kriterium benutzt, wobei der zuletzt aufgenommene Knoten als Ausgangspunkt für die weitere Knotensuche dient (Tiefensuche). Der daraus resultierende Harkplan ist in Abb. 24 zu sehen:

a	1	29	15	40	41	42	28	27	14	2	18	5	4	3	30	43	68	67	53	69	55	54	44	45
$s(a)$	1	15	14	54	54	28	27	14	2	3	5	4	3	3	43	55	67	53	53	55	54	54	45	46
a	46	34	8	9	23	12	11	13	25	24	38	26	39	51	50	37	49	35	21	7	6	19	20	33
$s(a)$	34	20	9	9	10	11	11	25	24	36	26	26	51	50	37	49	35	21	7	6	6	20	20	47
a	60	59	47	63	62	74	73	61	48	36	22	10	58	57	70	56	75	76	65	52	64	78	77	16
$s(a)$	59	47	48	62	48	73	61	48	36	22	10	10	57	56	56	56	76	77	52	64	77	77	77	16

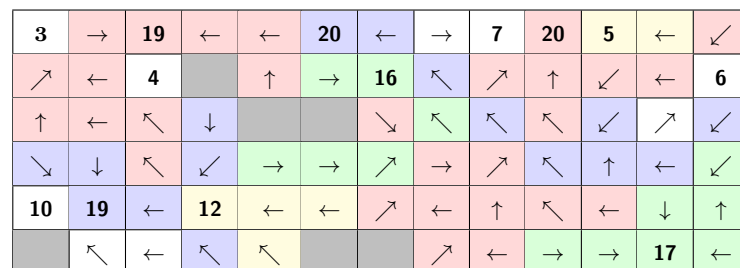


Abbildung 24: Harkplan durch huborientiertes Verfahren (TS) mit Minimalmengen-Kriterium am Beispiel

Es entstehen 13 Cluster mit teilweise gering angehäuften Laubharkmengen. Die Entsorgungskosten K des Harkplans nach *huborientiertem Konstruktionsverfahren mit Minimalmengen-Kriterium* belaufen sich bei Setzung $\alpha_H = 1$ auf $K(s) = HA_\Sigma(s) + WA_\Sigma(s) + TA_\Sigma(s) = 252 + 46 \cdot \alpha_W + 160 \cdot \alpha_T$ [ZE].

Durch Hubausrichtung in jedem der 13 Cluster (siehe Abb. 25) verringern sich die Gesamtkosten auf $K(s) = 220 + 57 \cdot \alpha_W + 160 \cdot \alpha_T$ [ZE], zumal $\alpha_W \leq \alpha_H$ angenommen werden kann.

a	1	14	2	5	18	4	27	29	30	43	41	40	42	28	15	3	19	8	9	22	12	11	13	25
$s(a)$	1	2	3	4	4	3	15	15	43	55	54	54	28	15	3	3	20	9	9	10	11	11	25	24
a	24	37	63	62	60	33	59	47	73	74	61	48	36	23	10	38	26	39	51	50	49	35	21	7
$s(a)$	10	49	62	48	48	47	47	48	61	61	48	36	23	10	10	26	26	51	50	49	35	21	7	6
a	6	44	45	46	34	20	58	57	68	67	53	69	55	54	70	56	75	76	65	78	52	64	77	16
$s(a)$	6	45	46	34	20	20	57	56	67	53	53	55	54	54	56	56	76	77	77	77	64	77	77	16

3	→	19	←	←	20	←	→	7	20	5	←	↘
↗	↗	4	↖	↖	→	16	↖	↗	↑	↖	←	6
↗	↑	↖	↓	↖	↖	↖	↖	↑	↖	↖	↖	↖
↘	↓	↖	↖	→	→	↗	→	↗	↖	↖	↖	↖
10	19	←	12	←	←	↗	↗	↑	↖	←	↓	↖
↖	↖	↖	↖	↖	↖	↖	↖	↑	→	→	17	←

Abbildung 25: Harkplan durch huborientiertes Verfahren (TS) mit Minimalmengen-Kriterium nach Hubausrichtung am Beispiel

Unter denselben Bedingungen wie oben sind die Entsorgungskosten K des Harkplans nach *huborientiertem Konstruktionsverfahren mit Maximalmengen-Kriterium* (siehe Abb. 26): $K(s) = 321 + 30 \cdot \alpha_W + 162 \cdot \alpha_T$ [ZE].

a	1	27	28	41	40	67	53	54	42	30	29	15	14	2	16	3	18	4	5	35	63	74	73	61
$s(a)$	1	28	41	40	53	53	54	54	30	29	15	14	2	16	3	3	4	5	19	49	62	73	61	62
a	62	49	37	50	64	78	77	65	52	51	39	38	26	13	12	11	23	22	21	7	8	9	10	25
$s(a)$	49	37	50	64	78	77	77	52	51	39	38	26	26	12	11	11	22	21	7	8	9	10	10	24
a	24	36	48	47	33	19	20	6	44	58	57	69	68	55	43	56	60	59	45	46	34	76	75	70
$s(a)$	36	48	47	33	19	20	6	6	45	57	43	68	55	43	56	56	59	45	46	34	34	75	75	70

3	↘	20	→	↘	20	→	→	→	20	6	←	←
↗	←	↑	↖	↖	→	↖	↖	←	←	↖	←	14
→	↓	↖	←	↖	↖	↖	11	↖	↖	↓	↗	←
↓	←	↗	↓	→	→	↗	↖	←	↗	↖	↗	←
→	19	↗	20	↖	←	↖	←	→	↑	←	↖	↑
↖	↖	↑	←	2	↖	↖	↖	←	4	←	19	←

Abbildung 26: Harkplan durch huborientiertes Verfahren (TS) mit Maximalmengen-Kriterium nach Hubausrichtung am Beispiel

Durch Hubausrichtung in jedem der 12 Cluster (siehe Abb. 27) verringern sich die Gesamtkosten auf $K(s) = 190 + 51 \cdot \alpha_W + 162 \cdot \alpha_T$ [ZE] deutlich, wenn auch hier wieder $\alpha_W \leq \alpha_H = 1$ angenommen wird.

a	1	14	2	4	5	7	8	21	9	22	23	10	25	24	36	48	47	33	19	18	20	6	30	16
$s(a)$	1	2	3	5	6	8	9	9	10	10	10	10	24	36	48	47	33	19	6	6	6	6	16	3

a	27	28	40	41	42	29	15	3	43	44	45	58	57	55	68	69	56	67	53	54	59	60	46	34
$s(a)$	40	40	54	54	29	15	3	3	56	45	46	57	56	56	56	56	56	54	54	54	46	46	34	34

a	35	49	37	50	64	65	51	39	13	12	11	52	38	26	74	76	75	73	61	62	63	78	77	70
$s(a)$	49	63	50	64	77	51	38	26	12	11	11	38	26	26	62	75	75	61	62	63	77	77	77	70

3	→	20	→	→	20	→	→	→	20	6	←	←
↗	↗	↑		↗	↑	↖	↗	↗	↑	↖	←	14
↓	↙	↖	↖			↖	11	↘	↙	↓	↗	↑
↘	↓	↑	↓	→	→	↗	↖	←	↘	↘	↑	↖
→	19	→	20	←	←	↑	↖	→	→	↘	↓	↖
	↑	↗	↑	2			↗	↗	4	←	19	←

Abbildung 27: Harkplan durch huborientiertes Verfahren (TS) mit Maximalmengen-Kriterium am Beispiel

C.3.2 Huborientiertes Konstruktionsverfahren via Breitensuche

Der folgende Harkplan (Abb. 28) entsteht durch das *huborientierte Konstruktionsverfahren via Breitensuche* (BS).⁸⁸

a	2	4	5	18	19	8	9	11	22	23	24	10	12	13	25	39	50	51	52	64	63	65	76	78
$s(a)$	3	3	6	6	6	9	10	10	10	10	10	10	26	26	26	26	64	64	64	77	77	77	77	77

a	77	62	75	61	47	33	45	44	42	43	55	40	41	53	67	54	68	57	69	70	56	28	29	30
$s(a)$	77	75	75	47	46	45	58	56	56	56	54	54	54	54	54	54	56	56	56	56	56	16	16	16

a	16	14	1	27	15	3	59	34	20	21	7	6	35	36	48	49	37	38	26	74	73	60	46	58
$s(a)$	3	1	1	15	3	3	58	20	6	7	6	6	49	37	49	37	38	26	26	60	60	46	58	58

5	→	20	←	→	20	←	→	→	20	←	↘	↓
↑	↗	↑	↖	↗	↑	↖	↖	↗	↑	↖	→	20
↗	↗	↑	↖	↖	↖	↖	↖	↖	→	→	↗	↑
↘	↓	↘	↓	↖	↓	↖	←	→	↗	↘	↓	↖
→	17	←	20	←	9	←	↖	↖	↓	↘	↓	↖
↖	↑	↗	↑	↖	↖	↖	↑	↖	3	→	20	←

Abbildung 28: Harkplan durch huborientiertes Verfahren (BS) am Beispiel

Die Entsorgungskosten setzen sich danach aus einem relativ geringen Harkaufwand $HA_{\Sigma}(s) = 163 \cdot \alpha_H$ [ZE], einem höheren Wegeaufwand $WA_{\Sigma}(s) = 68 \cdot \alpha_W$ und dem Transportaufwand $TA_{\Sigma}(s) = 130 \cdot \alpha_T$ zusammen. Bei Standardsetzung $\alpha_H = 1$ ergibt sich: $K(s) = 163 + 68 \cdot \alpha_W + 130 \cdot \alpha_T$ [ZE]. Durch Huborientierung lässt sich der Harkaufwand nicht mehr verringern.

C.4 Abschließende Erläuterungen zu den Beispielrechnungen

In der folgenden Tabelle sind die Beispielergebnisse zu den verschiedenen Harkstrategien zusammengefasst (Tab. 10). Die Prozentangaben stellen das jeweilige Zeitkostenergebnis in Relation zur besten ermittelten Lösung (rot markiert).

Es zeigt sich zwar ein „Gütetrend“, aber keine endgültige, von den Parametern α_H , α_W und α_T unabhängige Dominanz. Aufgrund der Annahme $\alpha_W \leq \alpha_H$, hängt die Lösungsgüte im Wesentlichen vom Verhältnis $\alpha_H \div \alpha_T$ zwischen Hark- und Transportparameter ab.

⁸⁸ Hierzu siehe auch Algorithmus 13 im Anhang B.

Tabelle 10: Ergebnisvergleich für Beispielgarten

Verfahren	$K(s) = HA_{\Sigma} + WA_{\Sigma} + TA_{\Sigma}$	$\alpha_H = \alpha_W = \alpha_T = 1$	$\alpha_H = 1; \alpha_W = 0,5; \alpha_T = 2$
Einfaches Zickzack	$517 \cdot \alpha_H + 14 \cdot \alpha_W + 116 \cdot \alpha_T$	647 [179,2%]	756,0 [165,4%]
plus Harkausrichtung	$254 \cdot \alpha_H + 47 \cdot \alpha_W + 114 \cdot \alpha_T$	415 [115,0%]	505,5 [110,6%]
Unterteiltes Zickzack	$570 \cdot \alpha_H + 19 \cdot \alpha_W + 134 \cdot \alpha_T$	723 [200,3%]	847,5 [185,4%]
plus Harkausrichtung	$174 \cdot \alpha_H + 68 \cdot \alpha_W + 144 \cdot \alpha_T$	386 [106,9%]	496,0 [108,5%]
Clusterorient. V. (NW)	$197 \cdot \alpha_H + 65 \cdot \alpha_W + 126 \cdot \alpha_T$	388 [107,5%]	471,5 [103,2%]
Huborient. V. (TS/Min)	$252 \cdot \alpha_H + 46 \cdot \alpha_W + 160 \cdot \alpha_T$	458 [126,9%]	593,5 [129,9%]
plus Harkausrichtung	$220 \cdot \alpha_H + 57 \cdot \alpha_W + 160 \cdot \alpha_T$	437 [121,1%]	567,0 [124,1%]
Huborient. V. (TS/Max)	$321 \cdot \alpha_H + 30 \cdot \alpha_W + 162 \cdot \alpha_T$	513 [142,1%]	660,0 [144,4%]
plus Harkausrichtung	$190 \cdot \alpha_H + 51 \cdot \alpha_W + 162 \cdot \alpha_T$	403 [111,6%]	539,5 [118,1%]
Huborient. V. (BS)	$163 \cdot \alpha_H + 68 \cdot \alpha_W + 130 \cdot \alpha_T$	361 [100,0%]	457,0 [100,0%]

Die obigen Angaben zum jeweiligen Transportaufwand sind unter der Voraussetzung entstanden, dass ausschließlich Pendeltouren zwischen den Hubfeldern und dem Kompostfeld stattfinden, zu Ungunsten der Harkpläne mit größerer Anzahl an Hubs. Es ergeben sich allerdings Einsparungsmöglichkeiten beim Transportaufwand durch Einbeziehung von Sammeltouren, die mit Hilfe der heuristischen Savingsmethode zur Lösung von Tourenplanungsproblemen⁸⁹ ermittelt sind, wodurch sich der Transportaufwand erheblich relativiert. Allerdings sind bei den folgenden Angaben keine zusätzlichen Rüst- oder Aufladekosten berücksichtigt (d.h. $\gamma = \sigma = 0$), wodurch die Einsparungen als „großzügig“ anzusehen sind:

Tabelle 11: Einsparungsmöglichkeiten durch Sammeltouren

Verfahren	Harkplan	Einsparung [ZE]
Einfaches Zickzack	Abb. 19	8
plus Harkausrichtung	Abb. 20	11
Unterteiltes Zickzack	Abb. 21	32
plus Harkausrichtung	Abb. 22	27
Clusterorient. V. (NW)	Abb. 23	6
Huborient. V. (TS/Min)	Abb. 24/25	47
Huborient. V. (TS/Max)	Abb. 26/ 27	46
Huborient. V. (BS)	Abb. 28	15

⁸⁹ Hierzu siehe u.a. [1], [3].

Zwecks Vergleichsmöglichkeit mit den Werten aus Tab. 10 sind die Aufwände aller Harkpläne zum obigen Beispiel unter Einsatz von Sammeltouren in der folgenden Tabelle angegeben:

Tabelle 12: Ergebnisvergleich für Beispielgarten unter Berücksichtigung von Sammeltouren

Verfahren	$K(s) = HA_{\Sigma} + WA_{\Sigma} + TA_{\Sigma}$	$\alpha_H = \alpha_W = \alpha_T = 1$	$\alpha_H = 1; \alpha_W = 0,5; \alpha_T = 2$
Einfaches Zickzack	$517 \cdot \alpha_H + 14 \cdot \alpha_W + 108 \cdot \alpha_T$	639 [184,7%]	740,0 [173,3%]
plus Harkausrichtung	$254 \cdot \alpha_H + 47 \cdot \alpha_W + 103 \cdot \alpha_T$	404 [116,8%]	483,5 [113,2%]
Unterteiltes Zickzack	$570 \cdot \alpha_H + 19 \cdot \alpha_W + 102 \cdot \alpha_T$	691 [199,7%]	783,5 [183,5%]
plus Harkausrichtung	$174 \cdot \alpha_H + 68 \cdot \alpha_W + 117 \cdot \alpha_T$	359 [103,8%]	442,0 [103,5%]
Clusterorient. V. (NW)	$197 \cdot \alpha_H + 65 \cdot \alpha_W + 120 \cdot \alpha_T$	382 [110,4%]	469,5 [110,0%]
Huborient. V. (TS/Min)	$252 \cdot \alpha_H + 46 \cdot \alpha_W + 113 \cdot \alpha_T$	411 [118,8%]	501,0 [117,3%]
plus Harkausrichtung	$220 \cdot \alpha_H + 57 \cdot \alpha_W + 113 \cdot \alpha_T$	390 [112,7%]	474,5 [111,1%]
Huborient. V. (TS/Max)	$321 \cdot \alpha_H + 30 \cdot \alpha_W + 116 \cdot \alpha_T$	467 [135,0%]	568,0 [133,0%]
plus Harkausrichtung	$190 \cdot \alpha_H + 51 \cdot \alpha_W + 116 \cdot \alpha_T$	355 [102,6%]	446,5 [104,6%]
Huborient. V. (BS)	$163 \cdot \alpha_H + 68 \cdot \alpha_W + 115 \cdot \alpha_T$	346 [100,0%]	427,0 [100,0%]

Diese Ergebnisse sind, da sie sich auf ein spezielles Beispiel beziehen, natürlich nicht repräsentativ. Allgemeinere Effizienzaussagen entnehme man dem Abschn. ??.

Abschließend sei noch angemerkt, dass bei den obigen Beispielrechnungen der Einfachheit halber keine Sonderbehandlung der Leerfelder (vgl. 3.1.3) vorgenommen wurde.

Autor



Prof. Dr. rer. pol. Hermann-Josef Kruse

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

hkruse@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.

Kontakt Daten

Prof. Dr. Hermann-Josef Kruse

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

Interaktion 1

33619 Bielefeld

Telefon +49.521.106-7411

Telefax +49.521.106-7176

hkruse@fh-bielefeld.de

Raum D 229